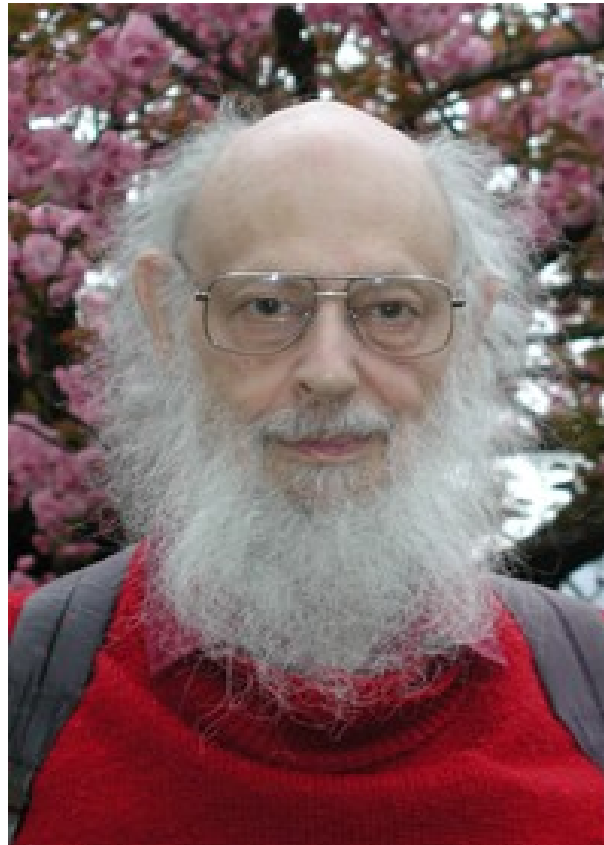


Универсален изкуствен разум

Машинно обучение (въведение)



Рей Соломонов (Ray Solomonoff) – 1926 - 2009
• <http://research.twenkid.com> Todor Arnaudov, Twenkid Research

Съдържание

- Какво е машинно обучение? Приложение.
- Защо е възможно?
- Вериги на Марков.
- Мрежи на Бейс. Скрити модели на Марков.
- Йерархични мрежи на Бейс и йерархични скрити модели на Марков.
- Принципи на алгоритмите на Витерби и Баум-Уелч (Expectation-Maximization).

Какво е машинно обучение?

- Алгоритми за изграждане на приспособяващо се поведение въз основа на модели, открити в данните.
- Математически модели за класификация и за предсказване на бъдещите данни въз основа на миналите и/или примери.
- Алгоритми за разпознаване на шаблони (pattern-recognition).
- **Приложение:** статистическата обработка на естествен език - машинен превод (Google Translate), разпознаване на реч, образи, ръкописен текст; извличане на информация (data mining), алгоритъм на Витерби за корекция на грешки в съобщителен канал и др.

Как е възможно?

- Представете си света като съвкупност от случайни величини и случайни събития, които се случват с определена вероятност.
- Във Вселената, съответно в данните обаче има определено количество зависимости и подреденост.
- **Редът и предсказуемостта** са обусловени от **излишеството** на информация и зависимостите (корелациите) между случайните величини и събития, защото **явленията всъщност не са напълно случайни и независими.**

Как е възможно?

- Събитията могат да се класифицират във вероятностни разпределения, по които могат да се правят предположения за други събития и случайни величини от подобен клас (индукция, от частното към общото).
- **В данни, породени от живи и разумни същества има изявени зависимости и предсказуемост.**

Статистическа вероятност

A 1

Б 2

В 3

Г 4

Д 5

A 1

В 3

Г 4

A 1

Б 2

В 3

Г 4

Д 6

A ? P(1) = 1

Д ? P(5) = 0.5

P(6) = 0.5

В

Г

? P(Д) = 2/3

P(А) = 1/3

Събитие?

- Формално – избор на елемент от множество „събития“.
- Обобщено - промяна, нещо което може да се:

Различи от други и да се класифицира
Да се измери честота на случване
Да се сравни честота на случване

Статистическа вероятност (преговор)

- Още – емпирична вероятност
- Наблюдения (O - observations)
- Какви реални (осезаеми, наблюдаеми) събития се случват, колко често, с каква вероятност се случват.
- **Условна вероятност (формула на Бейс)** - каква е вероятността за случване на B, ако преди това се е случило A.
- Независими случайни величини

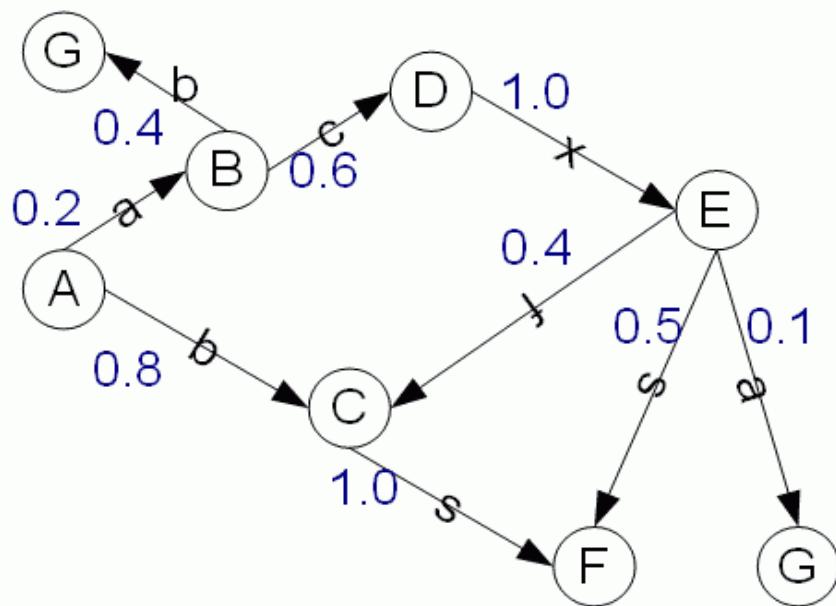
Вероятностна независимост

- Вероятността за случване на събитие В не зависи от това дали преди това се е случило или не събитие А. (След случване на А, В не се случва нито по-често, нито по-рядко.)
- Хвърлени зарове, монети и т.н., последователно бъркане в различни съдове, от които се взема предмети - всеки следващ опит е независим.

$$\Pr(R \cap B | G) = \Pr(R | G) \Pr(B | G),$$

Вериги на Марков - представяне

- Претеглен насочен граф (DAG) **без цикли**.
- Теглата са вероятности за преход.
- Сборът на коефициентите на ребрата, излизащи от всеки възел е 1.



Вериги на Марков - определение

- X_1, X_2, X_3 е редица от случайни величини, които имат свойството на Марков.
- X_1, X_2, X_3, \dots - конкретни дискретни състояния, например на недетрминиран краен автомат
- x_1, x_2, x_3, \dots - вероятност да се премине в това състояние в дадената стъпка
- **Свойство на Марков** – вероятността за преминаване в състояние X_{n+1} зависи единствено от това кое е предходното състояние.

$$\Pr(X_{n+1} = x | X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \Pr(X_{n+1} = x | X_n = x_n).$$

Вериги на Марков

- Описание на процеси, които се променят в дискретно зададено време, на тактове.
- *Вериги на Марков от m -ти (емти) ред* – новото състояние зависи само от m предходни, и не повече от m .

$$\Pr(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_1 = x_1)$$

$$= \Pr(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \dots, X_{n-m} = x_{n-m}) \text{ for } n > m$$

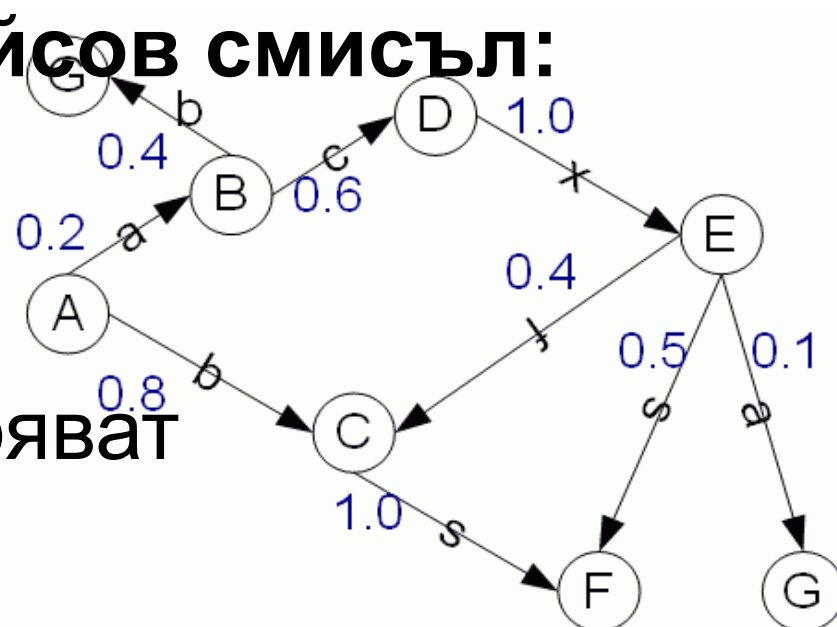
Процеси на Марков

- Като веригите на Марков, но в непрекъснатото време.

$$\begin{aligned} \Pr [X(t) = x(t) \mid X(s) = x(s), X(p_1) = x(p_1), X(p_2) = x(p_2), \dots] = \\ = \Pr [X(t) = x(t) \mid X(s) = x(s)] \end{aligned}$$

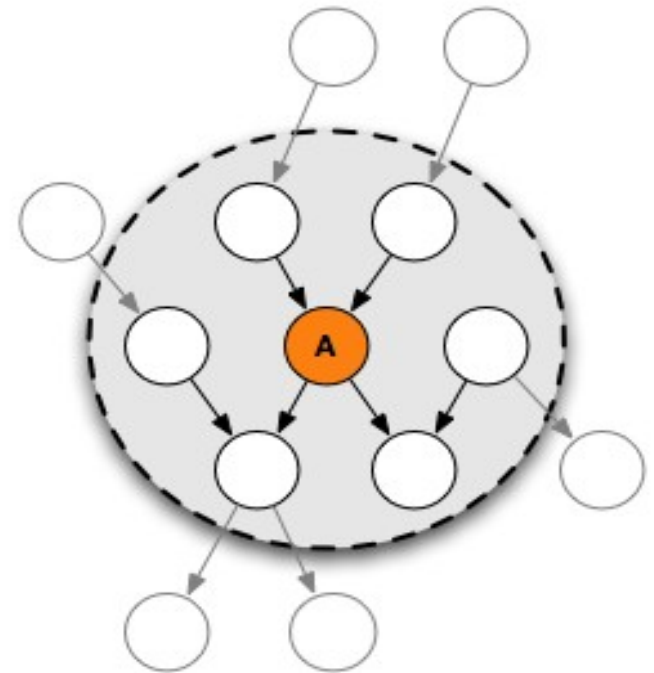
Мрежи на Бейс

- **Подобни на веригите на Марков – DAG** без цикли, съставен от възли (*случайни величини в Бейсов смисъл*) и вероятностни зависимости между тях
- **Случайни величини в Бейсов смисъл:**
 - наблюдения (стойности)
 - скрити променливи
 - неизвестни параметри
 - хипотези, които се проверяват
- **Априорна вероятност**
 - начална вероятност на възлите (после се настройва от опитите)



„Одеало“ на Марков ∂A

- Възел в мрежа на Бейс, неговите родители, наследници, и другите родители на наследниците му.
- „Одеалото“ е вероятно независимо от другата част на мрежата:
Вероятността за A , при услови B (извън одеалото) == A при условие случване на събитие от одеалото ∂A .



$$\Pr(A \mid \partial A, B) = \Pr(A \mid \partial A).$$

Скрити модели на Марков (НММ)

- Динамична мрежа на Бейс (най-проста форма)

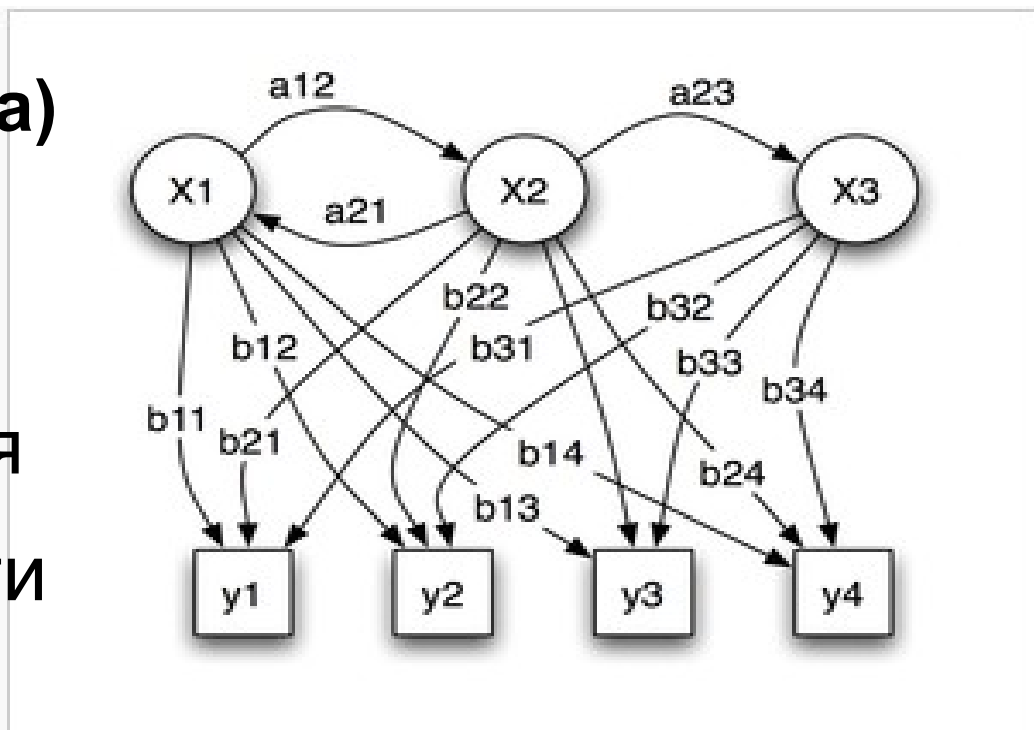
- x_1, x_2, x_3 – (скрити) състояния

- y_1, y_2, y_3 – наблюдения

- $a_{12}, a_{21} \dots$ - вероятности за преход между състоянията

- b_{11}, \dots - изходяща вероятност

- *Кодират последователности от случайни величини във времето – разпознаване на реч и др.*



Probabilistic parameters of a hidden Markov model (example)

x — states

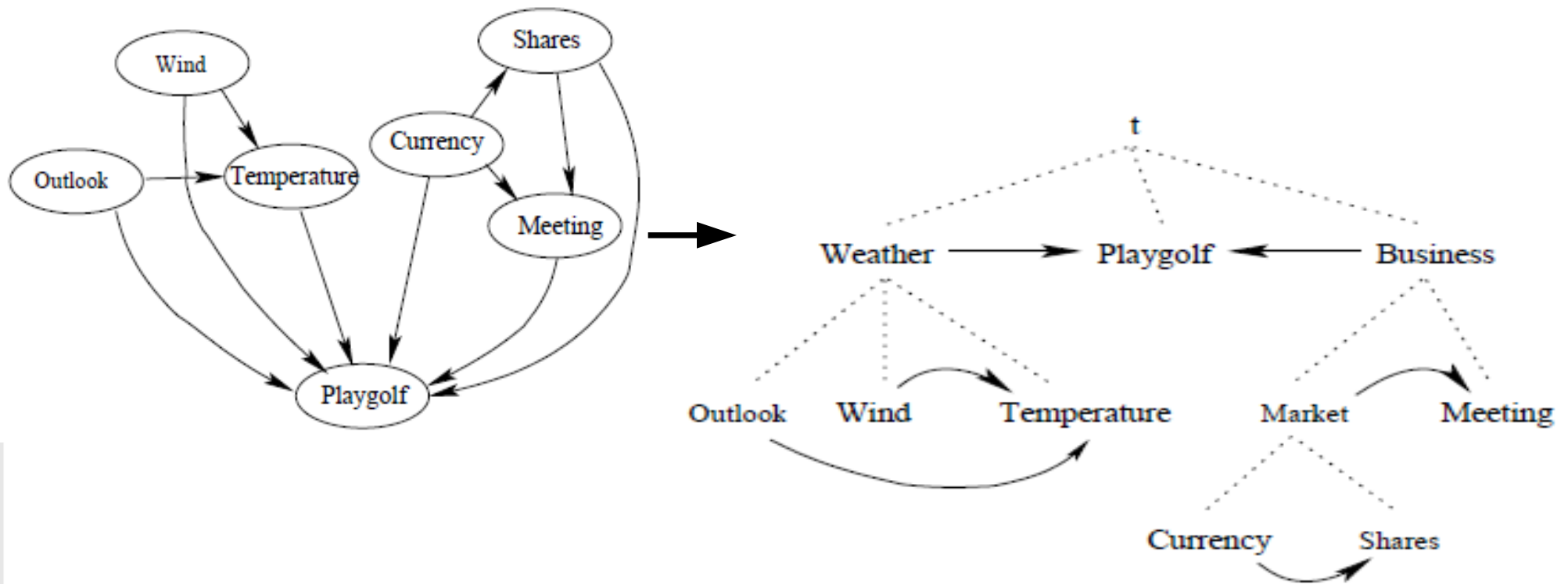
y — possible observations

a — state transition probabilities

b — output probabilities



Йерархични мрежи на Бейс

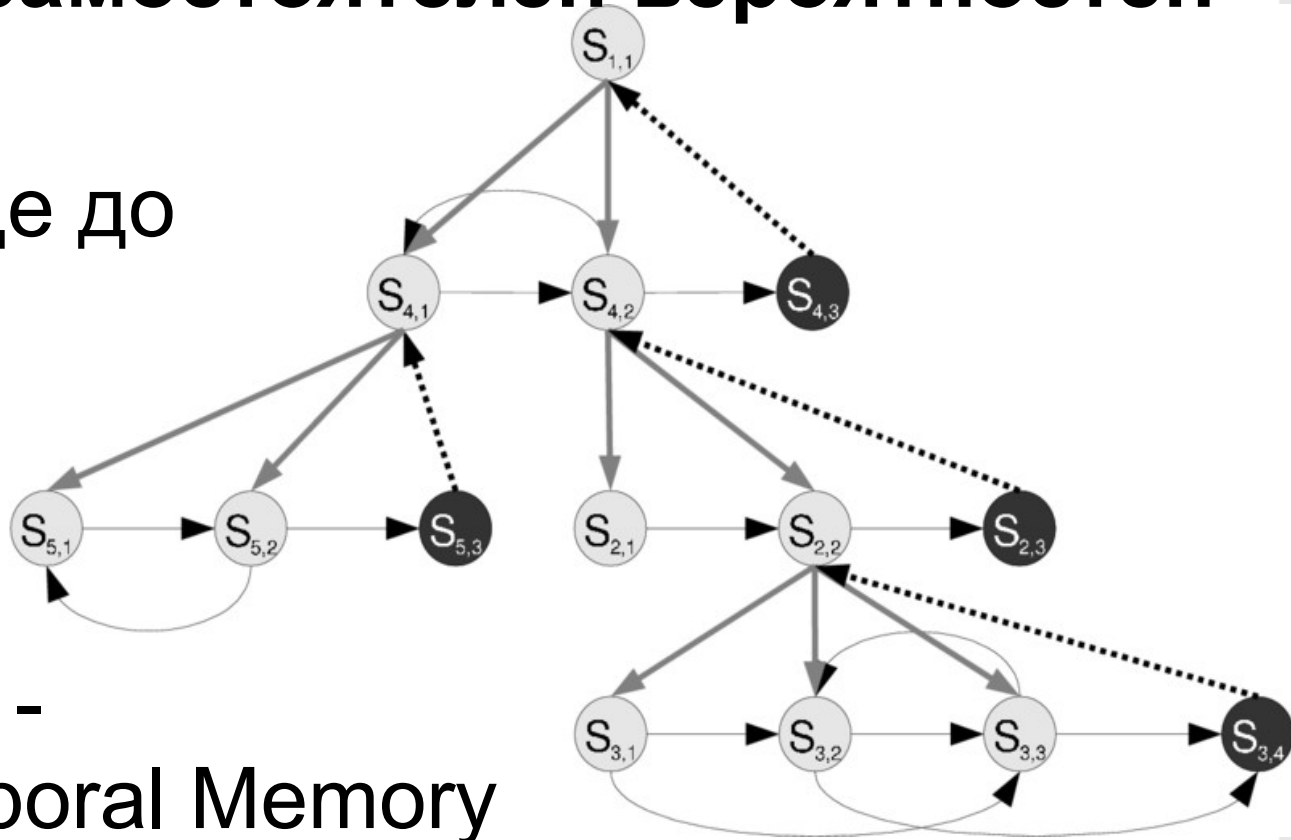


- По-ясно изразена структура и групиране между случайните величини.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.5107&rep=rep1&type=pdf>

Йерархични скрити модели на Марков (ННММ)

- **Всеки възел е самостоятелен вероятностен модел.**
- Може да се сведе до обикновен НММ
- По-ефективен за някои задачи
- Сходства с НТМ - Hierarchical Temporal Memory (виж лекцията за нея)



Алгоритъм на Витерби

- Намиране на най-вероятната последователност от състояния в скрит модел на Марков, която води до последователността от наблюдавани събития.
- Известни са само събитията (наблюдения).
- Търси се най-правдоподобна интерпретация на наблюденията.
- Andrew Viterbi 1967 – кодиране с шумопотискане

Алгоритъм на Витерби

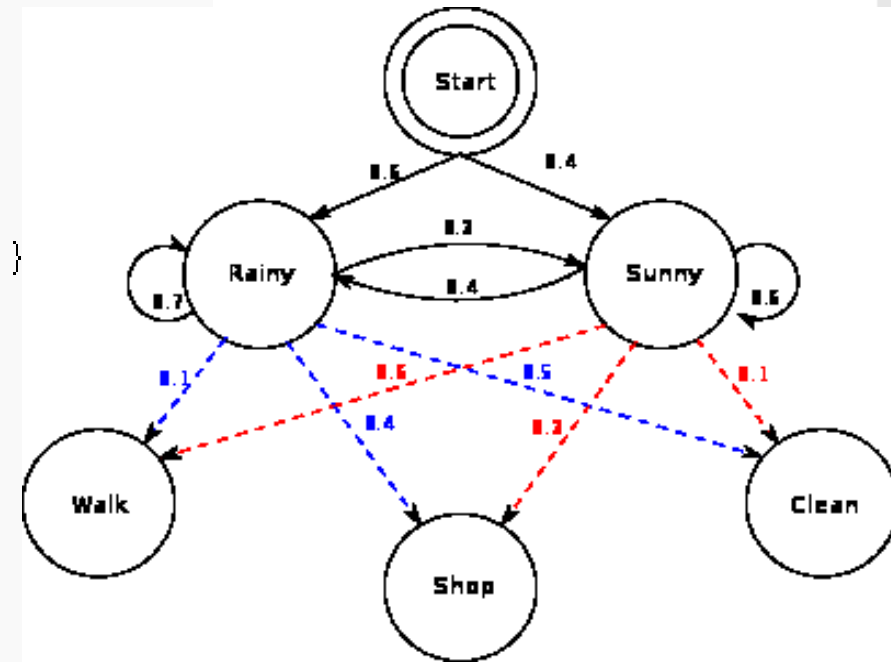
```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
    'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
    'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
}

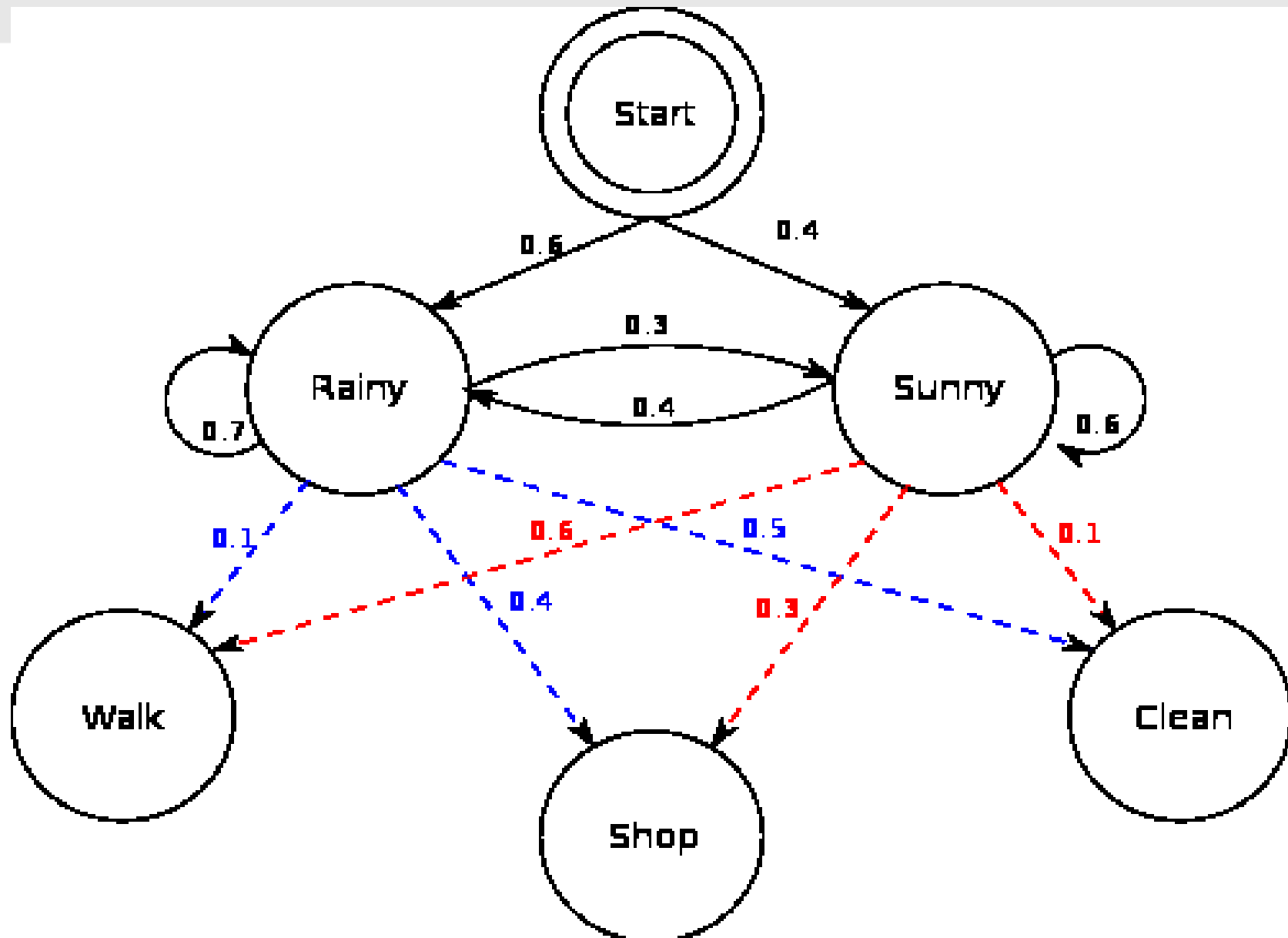
emission_probability = {
    'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
    'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
}
```



Иванчо се обажда от САЩ на Марийка всеки ден.



Алгоритъм на Витерби



Алгоритъм на Витерби

- (Скрити) Състояния (states).
- Наблюдения (observations).
- Вероятности за начално състояние (start probabilities).
- Вероятности за преходи между състояния (transition probabilities).
- Вероятности между състоянията и наблюденията (emission probabilities) – при определено наблюдение да се намираме в дадено състояние.
- „Оцелял път“ (survivor path) – предполагаем най-вероятен път от преходи.

Алгоритъм на Витерби (forward-viterbi)

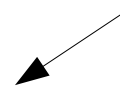
```
def forward_viterbi(obs, states, start_p, trans_p, emit_p):
    T = {}
    for state in states:
        ##          prob.          V. path  V. prob.
        T[state] = (start_p[state], [state], start_p[state])
    for output in obs:
        U = {}
        for next_state in states:
            total = 0
            argmax = None
            valmax = 0
            for source_state in states:
                (prob, v_path, v_prob) = T[source_state]
                p = emit_p[source_state][output] * trans_p[source_state][next_state]
                prob *= p
                v_prob *= p
                total += prob
                if v_prob > valmax:
                    argmax = v_path + [next_state]
                    valmax = v_prob
            U[next_state] = (total, argmax, valmax)
        T = U
    ## apply sum/max to the final states:
    total = 0
    argmax = None
    valmax = 0
    for state in states:
        (prob, v_path, v_prob) = T[state]
        total += prob
        if v_prob > valmax:
            argmax = v_path
            valmax = v_prob
    return (total, argmax, valmax)
```



Алгоритъм на Витерби (forward-viterbi)

```
return forward_viterbi(observations,  
                       states,  
                       start_probability,  
                       transition_probability,  
                       emission_probability)
```

*За първата стъпка
оптималната вероятност
съвпада с началната*



```
for state in states:
```

```
##          Probability  Viterbi path  Viterbi probability  
T[state] = (start_p[state], [state], start_p[state])
```

```
for output in obs:  
    U = {}  
    for next_state in states:  
        total = 0  
        argmax = None  
        valmax = 0  
        for source state in states:
```

Резултат

*Обхожда наблюденията,
след това състоянията и
техните съседни*

Алгоритъм на Витерби (forward-viterbi)

```
for source_state in states:
    (prob, v_path, v_prob) = T[source_state]
    p = emit_p[source_state][output] * trans_p[source_state][next_state]
    prob *= p
    v_prob *= p
    total += prob
    if v_prob > valmax:
        argmax = v_path + [next_state]
        valmax = v_prob
U[next_state] = (total, argmax, valmax)
```

Временен резултат

```
## apply sum/max to the final states:
total = 0
argmax = None
valmax = 0
for state in states:
    (prob, v_path, v_prob) = T[state]
    total += prob
    if v_prob > valmax:
        argmax = v_path
        valmax = v_prob
return (total, argmax, valmax)
```

Краино състояние (няма next_state като другите)

*Нагрупа вероятностите (умножава).
Ако новата вероятност > най-голямата досега – отбелязва пътя.*

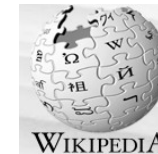
Алгоритъм на Витерби (forward-viterbi)

Наблюдение = ['walk', 'shop', 'clean']

$P(\text{observations}) = 0.033612$

Viterbi Path = ['Sunny', 'Rainy', 'Rainy', 'Rainy']

$P(\text{path}) = 0.00940$



- При дадения вероятностен модел на привички на Марийка, и разговорите на Иванчо с нея по телефона (че е била на **разходка, пазар, чистила**), най-вероятно времето е било: **слънчево, дъждовно, дъждовно, дъждовно**
- **Не е задължително реално да е било така.**
- **Вероятностите се донастройват от опита.**

Baum-Welch, Expectation-maximization

- *Оптимизиране на очакването*
- Итеративно търсене на най-вероятните параметрите на статистически модел, зависещи от ненаблюдавани скрити променливи (напр. в Скрит модел на Марков)
 1. Очакване $E \rightarrow E=?$ спрямо текущата оценка на скритите променливи.
 2. Максимизиране M – кои стойности на параметрите биха довели до максимална стойност на E на скритите променливи.
 3. GOTO 1

Обобщение

- Търсят се неяви причини, теория обясняваща пораждането на данните.
- Възприятията за света могат да се представят като **наблюдения, събития и вероятностни модели с явни и скрити състояния.**
- Възможно е да се правят предсказания и изводи от непълни данни с помощта на формални математически методи.

Допълнително четене

- Algorithmic Probability – Hutter, Legg, ...
http://www.scholarpedia.org/article/Algorithmic_probability
- Algorithmic Probability - Theory and Applications, Ray Solomonoff.
<http://world.std.com/~rjs/alp-theory-and-applications.pdf>
- **Introduction to Statistical Machine Learning** – видеолекция и слайдове от Маркус Хутер:
http://videlectures.net/mlss08au_hutter_isml/
- <http://research.twenkid.com>
- <http://artificial-mind.blogspot.com>
-
-