Session 4

- Naive Bayes
- Bayesian Networks
- Reinforcement Learning

Naive Bayes

- V = set of classes
- $a_1, \ldots, a_n$ = set of attributes
- $v = \text{argmax}_{v_j \in V} \; P(v_j | a_1, a_2, \ldots, a_n)$
- Apply Bayes rule
  $v = \text{argmax}_{v_j \in V} \; \frac{P(a_1, a_2, \ldots, a_n | v_j) \cdot P(v_j)}{P(a_1, a_2, \ldots, a_n)}$
- Remove constant factor
  $v = \text{argmax}_{v_j \in V} \; P(a_1, a_2, \ldots, a_n | v_j) \cdot P(v_j)$
- Assume:

  attribute values conditionally independent given class

  $v = \text{argmax}_{v_j \in V} \; P(v_j) \cdot \prod_i P(a_i | v_j)$

Naive Bayes for Text Classification

procedure naive_bayes_text(E: texts, V: classes, Voc: vocabulary)
    for all texts A_i in E:
        preprocess(A_i, Voc)
    estimate $P(v_j)$ and $P(w_k|v_j)$ for all w_k in Voc and v_j in V:
        N_j = number of texts of class j
        N = number of texts
        $P(v_j) = N_j/N$
        n_{kj} = number of occurrences of w_k in texts of class j
        n_j = number of words in class j (counting doubles)
        $P(w_k|v_j) = (n_{kj} + 1)/(n_j + |\text{Voc}|)$

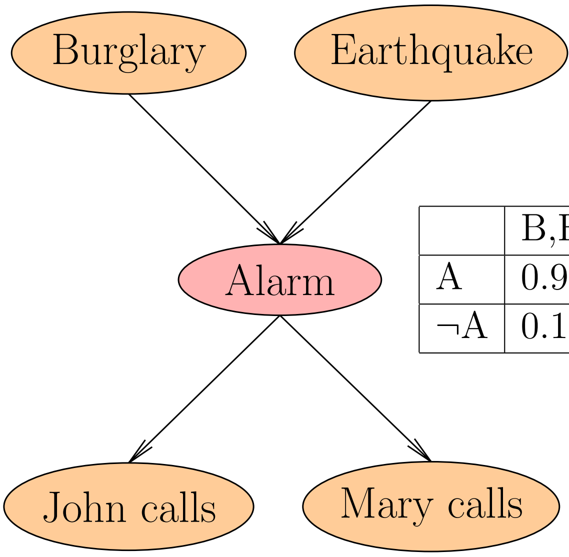
procedure classify_naive_bayes_text(A: text, Voc: vocabulary)
    preprocess(A, Voc)
    return $\text{argmax}_{v_j \in V} P(v_j) \times \prod_{w_i \in words(A)} P(w_i|v_j)$

procedure preprocess(A: text, Voc: vocabulary)
    transform each word in A to a standard form (stemming)
        e.g. *wait*, *waits*, *waited*, *waiting* $\Rightarrow$ *wait*
    remove from A all words/tokens that are not in Voc

Bayesian Networks

- Graph that represents joint probability distribution
- Node = attribute
- Edge = "direct influence" between two attributes
- Formally:

each node is conditionally independent of each of its non-descendants, given its parents



B	0.05
¬B	0.95

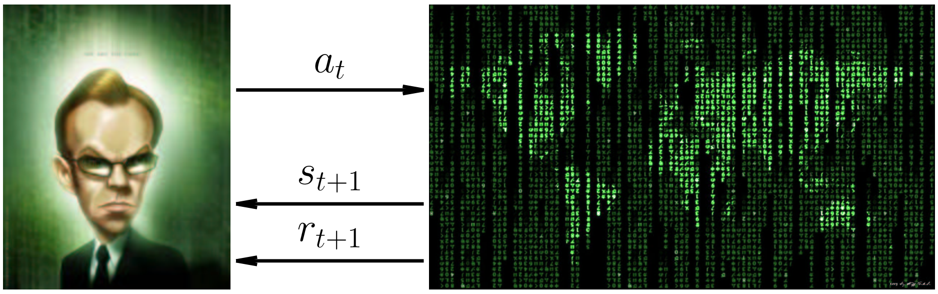
E	0.01
¬E	0.99

	B,E	B,¬E	¬B,E	¬B,¬E
A	0.9	0.8	0.4	0.01
¬A	0.1	0.2	0.6	0.99

	A	¬A
J	0.8	0.1
¬J	0.2	0.9

	A	¬A
M	0.9	0.2
¬M	0.1	0.8

Reinforcement Learning



- State space S
- Set of possible actions A
- Learn policy $\pi : S \to A$
- Unknown reward function $r : S \times A \to \mathbb{R}$
- Unknown next state function $\delta : S \times A \to S$

Value function: $V^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t \cdot r(s_t, a_t)$

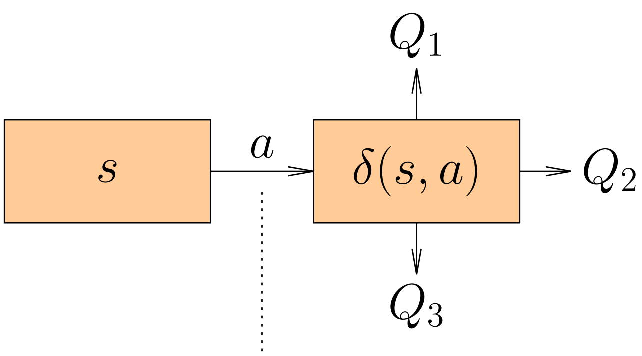
Discounted future rewards from following policy π from state s

Optimal policy: $\pi^*(s) = \text{argmax}_a \overbrace{r(s, a) + \gamma \cdot V^*(\delta(s, a))}^{Q(s,a)}$

Selects action a that maximises immediate reward and discounted future rewards

$V^*(s) = V^{\pi^*}(s) = \max_a Q(s, a)$

$Q(s, a) = r(s, a) + \gamma \cdot \max_{a'} Q(\delta(s, a), a')$



$Q(s, a) = r(s, a) + \gamma \cdot \max(Q_1, Q_2, Q_3)$