The Curse of Dimensionality

Guillaume Lozenguez

@imt-lille-douai.fr

IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille

System Difficulty

Directly correlated to the state space

The number of states: the Cartesian product of variable domains
(minus some unreachable states)

- **421 game:** 3 dice-6 at the horizon 3: $\left(3 \times 6^3 = 648\right)$
- **ZombieDice:**
  3 dice in 3 stocks and 2 scores: $\left(4^3 \times 7 \times 5 \times 4 \times 14 \times 4 = 501760\right)$
- **GO:** 3 possibilities over 19^2 positions $\left(3^{19 \times 19} \equiv 10^{172}\right)$

Then the branching:

The number of possible actions and actions' outcomes.

With a Classical 32-card game

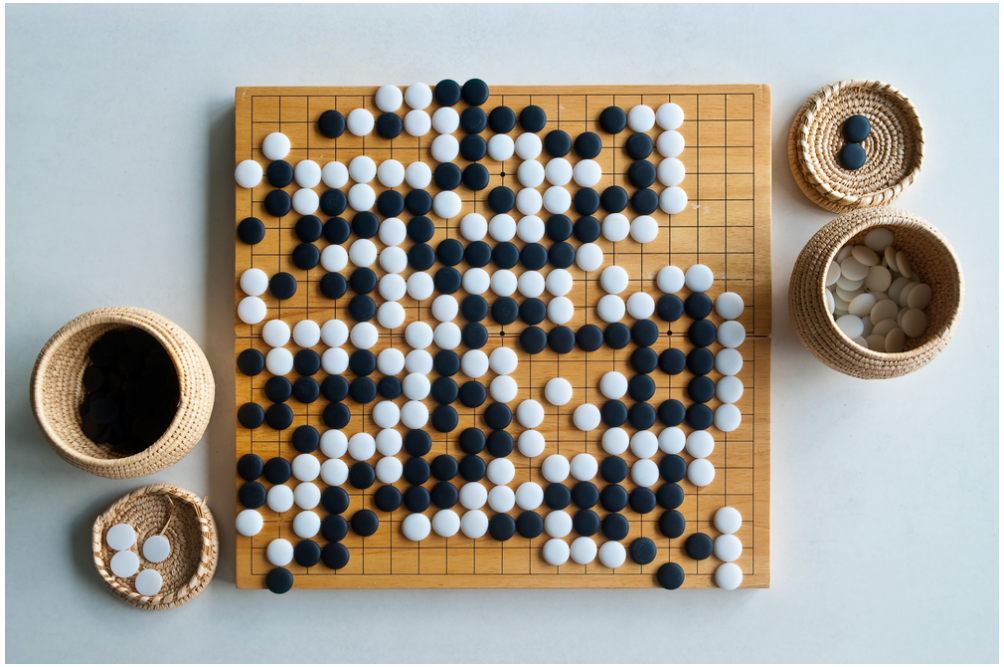
Possible distribution $32! = \quad 2.6 \times 10^{35}$

Human life: around 5×10^{7} seconds

Probability to play 2 times the same distribution in a human life is very close to 0

The notion of complexity (Go)

GO: 10^{170} positions, 10^{600} games (chess: 10^{120} games)



The notion of complexity (Go)

A classical 3 GHz computer: 3×10^9 op. per second
$\rightarrow$ 2.6×10^{14} op. a day $\rightarrow$ 10^{17} op. a year

Enumerating all games: $O(n)$ with $n = 10^{600}$: arround 10^{583} years.
$\rightarrow$ requires decomposed model and statistics...

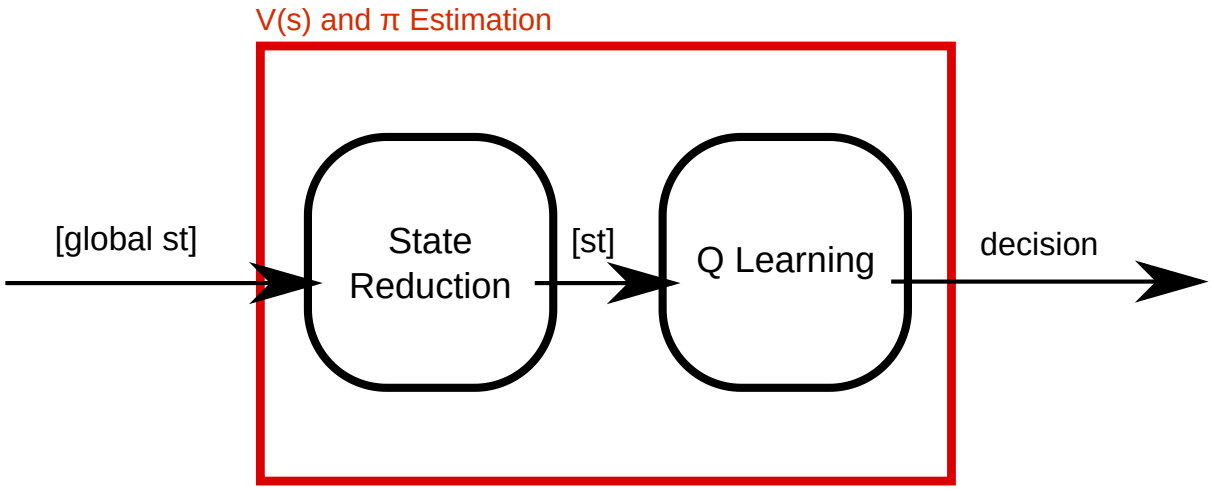
Sun life: arround 10^{30} years

The root problem: handle large systems

A first basic solution: reduce the state space.

State reduction in QLearning

Project the states in a space with reduced dimension



By mitigate the negative impact on the resulting built policy.

State reduction in QLearning

A classical unsupervised learning problem

- Group similar states :
  - close state (in the transition succession)
  - similar action outcome

Potentially: a supervised learning problem

- Group similar states :
  - similar Value (suppose to have some valued state)
  - similar action outcome

With a geometric approach

Principal Component Analysis (PCA)

Searching the hyper-plan that better separate the data, in a given dimension.

K-means

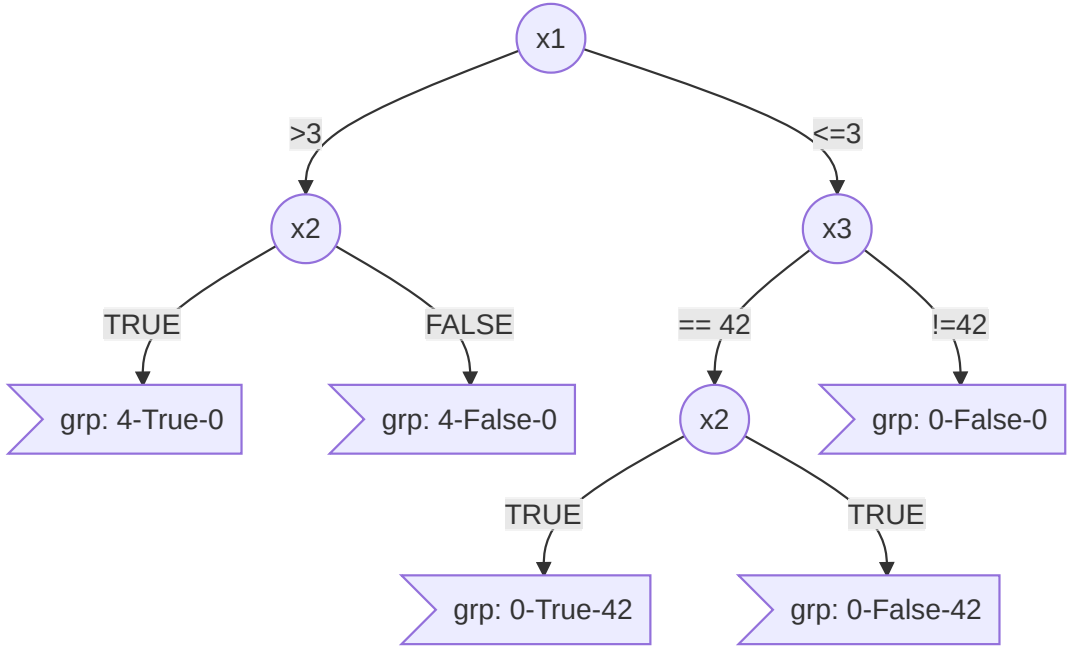
Searching the optimal k center positions that better group the data together.

- Work well with 'linear state transitions' and different states density.
- Suppose a data set (trace)

Based on state variable prevalence

Decision Tree

Nodes: variables ; **Edges:** assignment ; **leaf:** group of states



- Expert based or learned structure (ID3 algorithm)

Example: PCA + Discretization