# AI Review

# Outline

- Problem-solving
  - Classical search
  - Beyond classical search
  - Problem-Specific Search
- Machine Learning
  - Supervised Learning
  - Performance Evaluation
  - Unsupervised Learning
  - Automated Machine Learning
- Knowledge and Reasoning
  - Representing and Inference with logic
  - Representing and Inference with Uncertainty
  - Knowledge Graph and Recommender System

# Problem-solving

- Classical search(use a tree representation)
  - Uninformed Search
    - Breadth-first search (BFS)
    - Uniform-cost search (UCS)
    - Depth-first search (DFS)
    - Depth-limited search (DLS)
    - Iterative deepening search (IDS)
    - Bidirectional search
  - Heuristic (informed) Search
    - Greedy Best-first Search (Greedy Search)
    - A* Search
      - Admissible Heuristic
- Beyond classical search(generalize to other types of representation) -> solution space
  - “single-point” search framework
    - Local Search
    - Simulated Annealing
    - Tabu Search
    - Bayesian Optimization
  - Population-based search(Evolutionary Algorithms)
- Problem-Specific Search
  - CSP
  - Game

# Heuristic search

Uninformed search

no

Domain knowledge

yes

Heuristic search

evaluation function

f(x)

diff from BFS

node $x$ with the **lowest** $f(x)$ is **expanded first**

$f(x) = h(x)$

Greedy Best-first Search(Greedy search)

- $g(x)$: *cost so far to reach* $x$.
- $h(x)$: *estimated cost to goal from* $x$.
- $f(x)$: *estimated total cost of path through* $x$ *to goal.*

A*

$f(x) = g(x) + h(x)$

$h(x) = 0$

UCS

$f(x) = g(x)$

Heuristic function h is admissible if:
$\forall x \rightarrow h(x) \leqslant h^*(x)$, $h^*(x)$: the true cost from $x$ to goal.
A* with h is optimal if h is admissible.

For admissible h1 and h2, if $h2(x) \geqslant h1(x), \forall n$, then h2 dominates h1 and is more efficient for search.

# Population-based search(Evolutionary Algorithms)

Generate $P_0$ *with N initial solutions*

$i = 0$

Generate $P_{i+1}$ *base on* $P_i$:

1. Get fitness value for each individual in $P_i$
2. Generate M offsprings, each offspring generated by the following workflow:

   1.1 select 2 parents from $P_i$

   1.2 obtain 1 or 2 offsprings from crossover the 2 parents.

   1.3 apply mutation to the offspring with a small probability.

   1.4 calculate the fitness values of the offsprings.

3. Select **N individuals($\boldsymbol{P_{i+1}}$)** based on $P_i$ *and* M offsprings using some selection(replacement) strategy.

**How to generate a new individual**

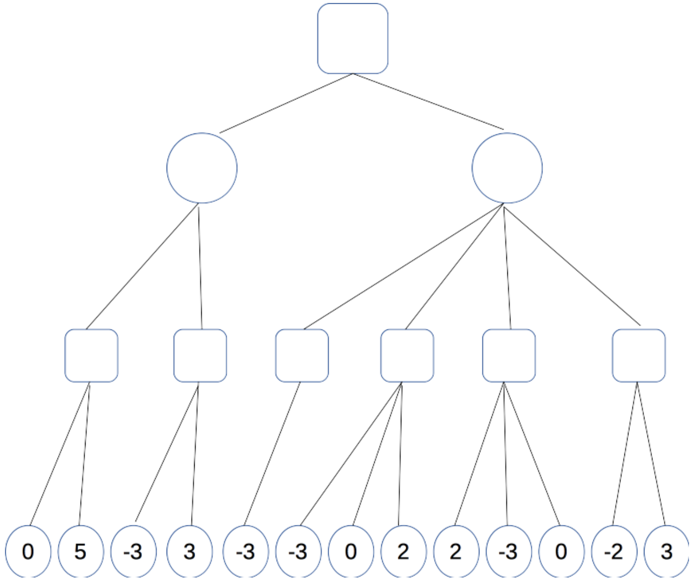
$i = i + 1$

Stop?

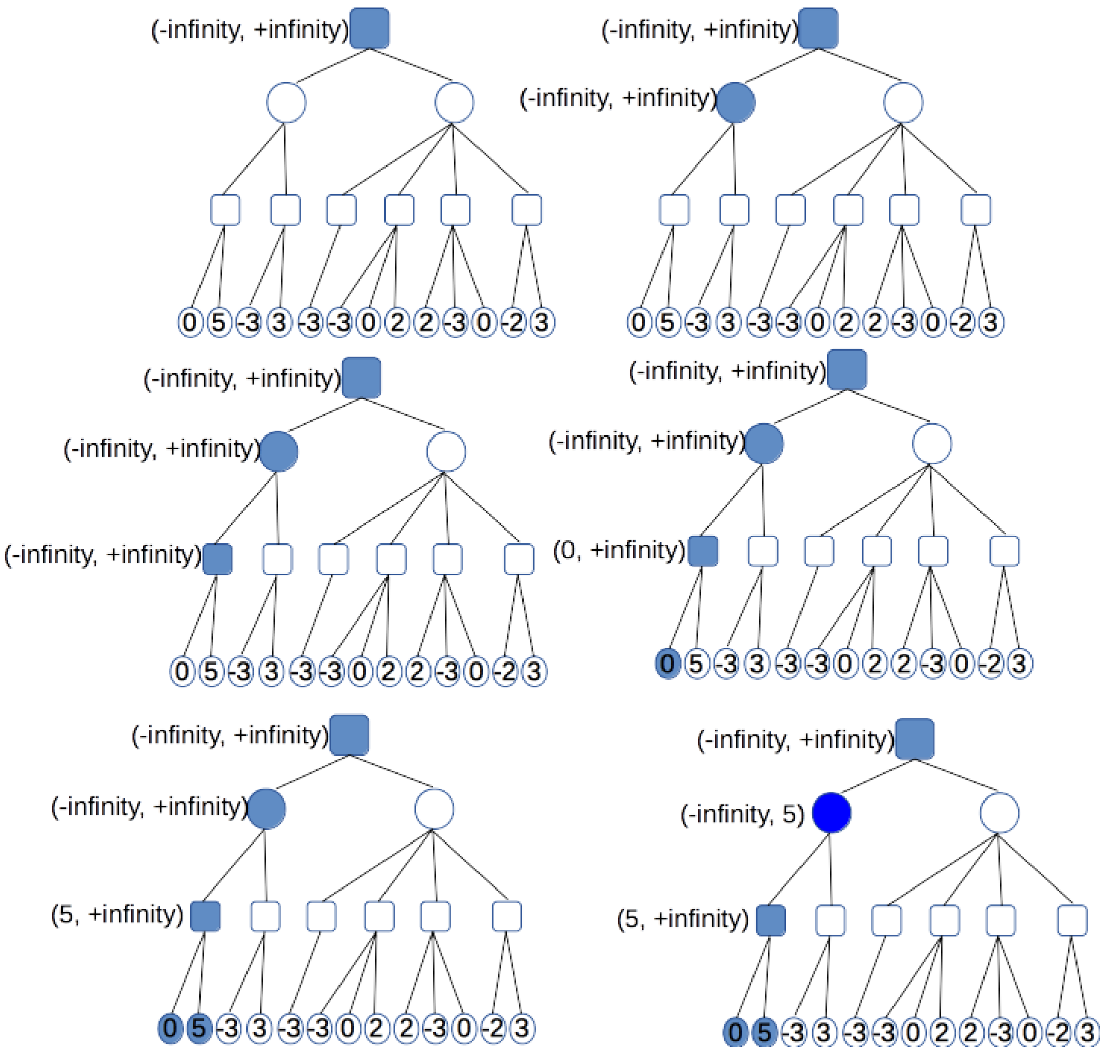
Return the best

# Game

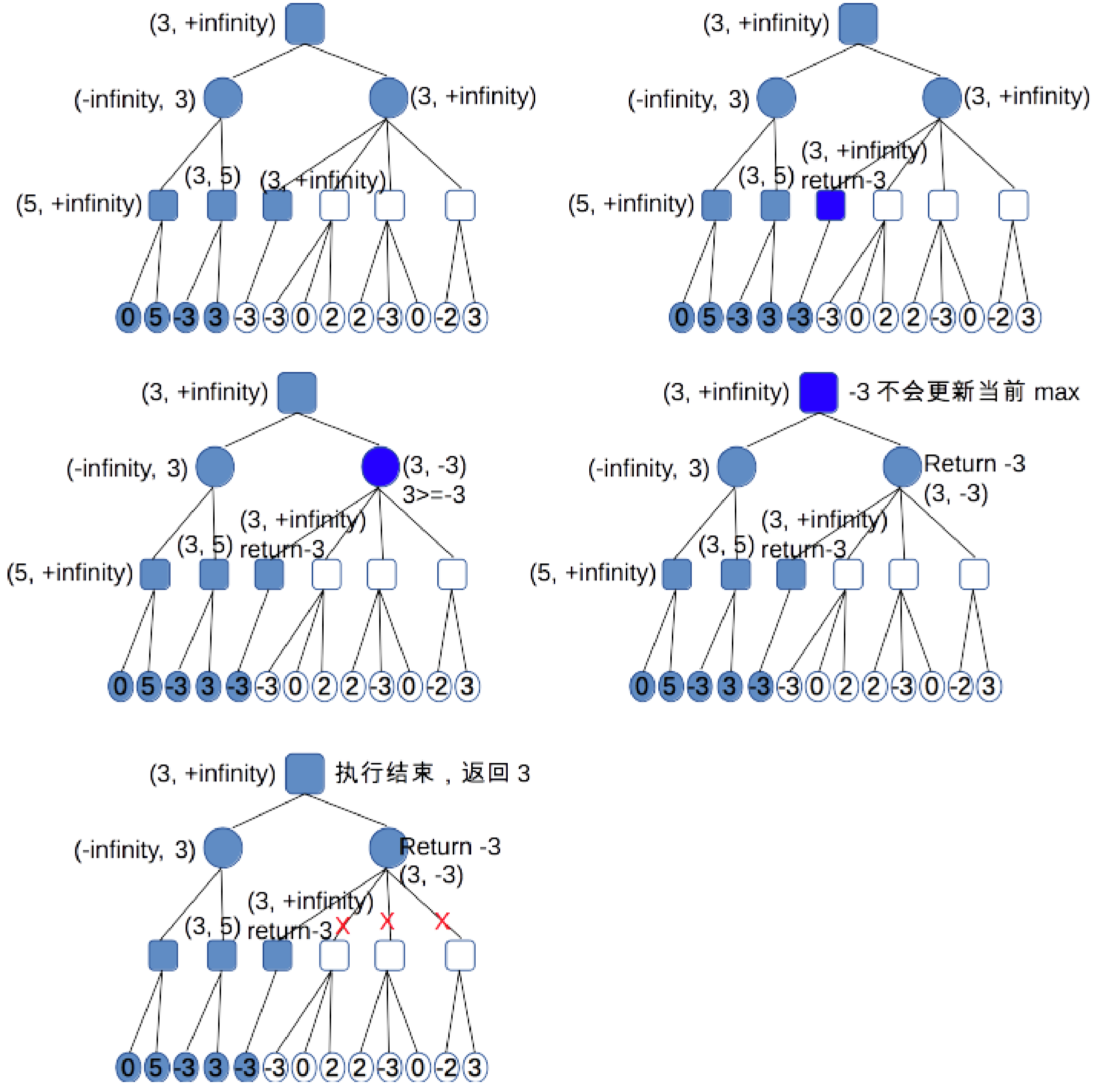
- For the following game tree, in which the numbers at the leaf nodes indicates their utility values, please applies Alpha-Beta pruning to prune unnecessary branches. Please directly labels the nodes with (Alpha, Beta) values, and put a “X” on branches that should be pruned.



# Ref answer:



(-infinity, +infinity)
(0, +infinity)
(5, +infinity)
(-infinity, 5)
0 5 -3 3 -3 -3 0 2 2 -3 0 -2 3

(3, +infinity)
(-infinity, 3)
(5, +infinity)
(3, 5)
(3, +infinity)
return-3
(3, -3)
3>=-3
-3 不会更新当前 max
Return -3
执行结束，返回 3
0 5 -3 3 -3 -3 0 2 2 -3 0 -2 3

# CSP

- Numbers 1-9 are needs to be filled in a 3 by 3 squares so that all the columns, rows and diagonals add up to 15.
  - Please formulate the above problem as a CSP.
  - Suppose Backtracking Search is applied to the problem, which variable will be first chosen to assign value (3 points), why?

# Ref answer:

- Please formulate the above problem as a CSP.

A constraint satisfaction problem consists of three components, $X, D, and\ C$ . For this problem:

| | | |
|---|---|---|
| $x_{11}$ | $x_{12}$ | $x_{13}$ |
| $x_{21}$ | $x_{22}$ | $x_{23}$ |
| $x_{31}$ | $x_{32}$ | $x_{33}$ |

$X\ = \{x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23}, x_{31}, x_{32}, x_{33}\}$

The domain of each variable is the set $D_i = \{1,2,3,4,5,6,7,8,9\}$

C: $\sum_{j=1}^{3} x_{ij} = 15\ \ i = 1,2,3$

$\sum_{i=1}^{3} x_{ij} = 15\ \ j = 1,2,3$

$\sum_{i=1}^{3} x_{ii} = 15$

$\sum_{i=1}^{3} x_{i(4-i)} = 15$

$x_{i_1 j_1} \neq x_{i_2 j_2}\ \ if\ i_1 \neq i_2, j_1\ \neq j_2$

- Suppose Backtracking Search is applied to the problem, which variable will be first chosen to assign value (3 points), why?

$x_{22}$, most constrained variable.

- Numbers 1-9 are needs to be filled in a 3 by 3 squares so that all the columns, rows and diagonals add up to 15.
  - Please formulate the above problem as a CSP.
  - Suppose Backtracking Search is applied to the problem, which variable will be first chosen to assign value (3 points), why?

How to solve the magic square problem using **Local Search?**

solution space?

**BFS?**

**DFS?**

…

# Machine Learning

- Supervised Learning
  - Support Vector Machines
  - Artificial Neural Networks
  - Decision Trees
- Performance Evaluation
  - Performance Metrics
  - Estimating the Generalization
- Unsupervised Learning
  - Clustering:  KMeans
  - Learning Low-Dimensional Representations: PVC, LLE
- Automated Machine Learning
  - The hyper-parameters of commonly used models
  - Tuning Hyper-parameters:  Grid Search, heuristic Search, …

# Confusion Matrix

- A binary classifier is applied to a dataset of 100 samples, and the following confusion matrix is obtained:

| | Predicted Positive | Predicted Negative |
|---|---|---|
| Actual Positive | 40 | 10 |
| Actual Negative | 20 | 30 |

Calculate the following metrics:

- Accuracy
- Precision
- Recall
- F1-measure

# Ref answer:

| | Predicted Positive | Predicted Negative |
|---|---|---|
| Positive | True Positive rate | False Negative rate |
| Negative | False Positive rate | True Negative rate |

$$accuracy = \frac{TPR \times N^{+} + TNR \times N^{-}}{N^{+} + N^{-}}$$

$$precision = \frac{TPR \times N^{+}}{TPR \times N^{+} + FPR \times N^{-}}$$

$$recall = \frac{TPR \times N^{+}}{TPR \times N^{+} + FNR \times N^{+}}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recl}$$

| | Predicted Positive | Predicted Negative |
|---|---|---|
| Actual Positive | 40 | 10 |
| Actual Negative | 20 | 30 |

Calculate the following metrics:

- Accuracy = (40+30)/100 = 0.7
- Precision = 40/(40+20) ≈ 0.67
- Recall = 40/(40+10) = 0.8
- F1-measure = 2*0.67*0.8/(0.67+0.8) ≈ 0.73

# ROC Analysis

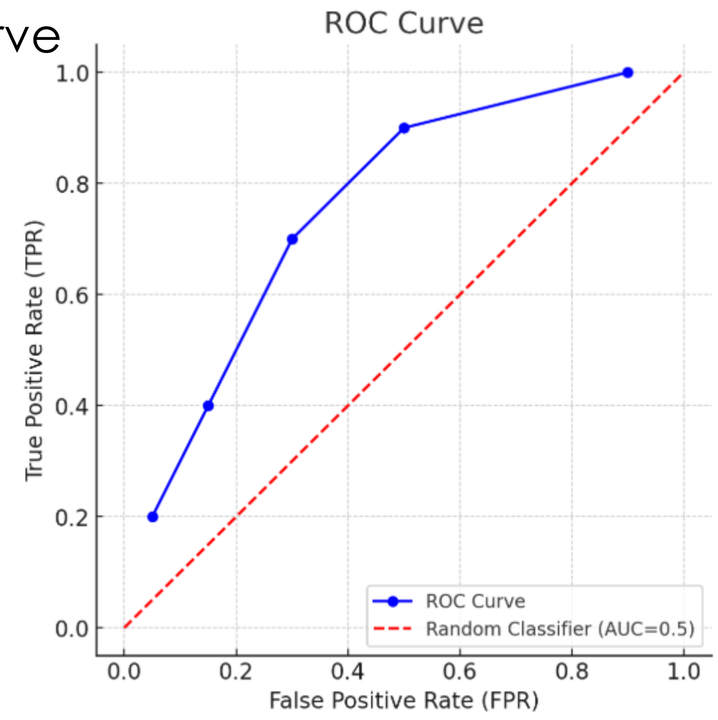
- The classifier from the above Question outputs a probability score for each sample. By varying the classification threshold, the following TPR (True Positive Rate) and FPR (False Positive Rate) values are recorded:

| Threshold | TPR | FPR |
|---|---|---|
| 0.1 | 1.00 | 0.90 |
| 0.3 | 0.90 | 0.50 |
| 0.5 | 0.70 | 0.30 |
| 0.7 | 0.40 | 0.15 |
| 0.9 | 0.20 | 0.05 |

(1)Plot the ROC curve using the given data points (FPR on the x-axis, TPR on the y-axis).
(2)Based on the ROC curve, explain how you would select the optimal threshold for this classifier.
(3)Define the AUC (Area Under Curve) and explain its significance in evaluating classifier performance.

# Ref answer:

(1)ROC Curve




(2) minimizing false positives while maximizing true positives, threshold = 0.3 or 0.5.
(3) **AUC (Area Under Curve)** is the area under the ROC curve measures the classifier's ability to distinguish between classes. A higher AUC indicates better classification performance. An AUC of 1.0 represents a perfect classifier, while an AUC of 0.5 corresponds to random guessing.

# Estimating the Generalization

- Generalization performance is a **random variable**.
- Split the data in hand into training and testing subsets.
  - Random Split
  - Cross-validation
  - Bootstrap
- Collecting the test performance for many times, calculate the average and standard deviation.
- Do statistical tests (check your textbook on statistics).

# Supervised Learning

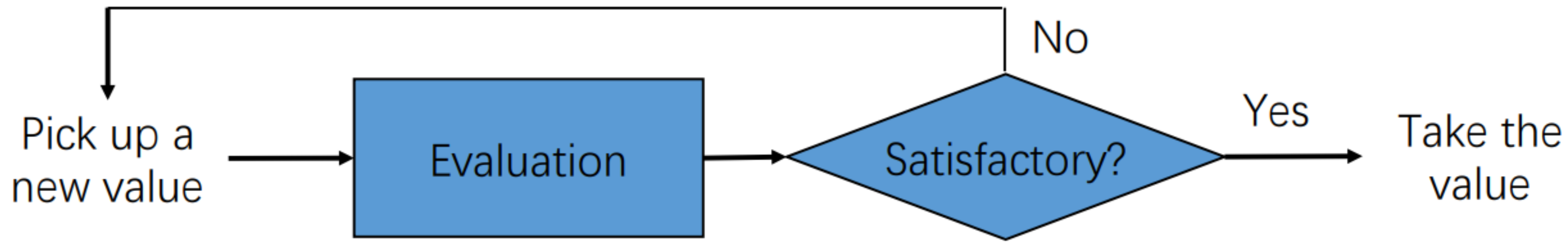
- Review Supervised learning Q&A .pptx

# Unsupervised learning

- Review Unsupervised learning.pptx

# Auto ML

- Supervised Learning
  - SVM: Kernel Parameters and Regularization Parameters
  - Neural Networks: Number of hidden nodes, activation functions, network architecture
  - Decision Tree: Branching factor, Height of the tree
- Unsupervised Learning
  - Clustering: Number of clusters



# Knowledge and Reasoning

- Representing and Inference with Logic
  - Represent Knowledge with Propositional Logic
  - Inference with Propositional Logic: CNF, SAT, DPLL, Resolution algorithm
  - Represent Knowledge with First-Order Logic: Syntax of FOL
  - Inference with FOL: reduce to propositional logic, Resolution， Chaining Algorithms
- Representing and Inference with Uncertainty
  - Bayesian Networks
  - Exact Inference with Bayesian Networks
  - Approximate Inference with Bayesian Networks
- Knowledge Graph and Recommender System
  - Recommender System: Content-based method, CF-based method, Hybrid method
  - KG and how to construct KG
  - Knowledge Graph Completion: Path-based method, Embedding-based method
  - KG-Based Recommender System

# Representing and Inference with Logic And Uncertainty

- Review Logic and Bayesian Questions Answer.pptx

# Knowledge Graph and Recommender System

- Review AI_lec08 and AI_lec13

- What is the key of a recommender system? What problems does it typically need to overcome?
- What is the Pearson Correlation Coefficient, and how is it used in collaborative filtering methods?

- Ref Answer：
- The core lies in the score function, which needs to efficiently generate recommendations in seconds. However, in many applications, the system must handle millions of users and items. There is always a trade-off between efficiency and accuracy.
- The Pearson Correlation Coefficient is a normalized measure of covariance that represents the correlation between two users or two items. In collaborative filtering, it is used to calculate the similarity between users or items. This similarity is then utilized to predict interaction probabilities.

$$c_{u_1u_2} = \frac{\sum_{i\in M}\left(r_{u_1,i} - \bar{r}_{u_1}\right)\left(r_{u_2,i} - \bar{r}_{u_2}\right)}{\sqrt{\sum_{i\in M}\left(r_{u_1,i} - \bar{r}_{u_1}\right)^2}\sqrt{\sum_{i\in M}\left(r_{u_2,i} - \bar{r}_{u_2}\right)^2}}$$

- An example of user’s Pearson Correlation Coefficient
- $M$: The item set
- $r_{u,i}$: Interaction record between user $u$ and item $i$
- $\bar{r}_u$: Mean value of all the interaction records of user $u$

- A user/item is represented by a vector that consists of all the correlation between itself and all the users/items
- Interaction probability of user $u$ and item $i$ can be calculated by a function $f(u,i)$, some simple example of $f$:
  - $f(u,i) = \sum_{u' \in U} c_{u,u'} r_{u',i}$, $U$ is user set, $c_{u,u'}$ is the correlation coefficient between $u$ and $u'$, $r_{u',i}$ is the interaction record of $u'$ and $i$.
  - $f(u,i) = \sum_{i' \in M} c_{i,i'} r_{u,i'}$, $M$ is item set, $c_{i,i'}$ is the correlation coefficient between $i$ and $i'$, $r_{u,i'}$ is the interaction record of $u$ and $i'$.