REINFORCEMENT LEARNING & KNOWLEDGE IN LEARNING

Artificial Intelligence

GROUP 8

1.REINFORCEMENT LEARNING

INTRODUCTION

Phương pháp học củng cố trong trí tuệ nhân tạo, nơi các tác nhân học từ phản hồi về hành vi của mình trong môi trường mà không cần có sự hướng dẫn từ một giáo viên hay tập dữ liệu đã được gán nhãn.

Thay vào đó, các tác nhân chỉ nhận được phản hồi sau khi hoàn thành một chuỗi hành động, thường là ở trạng thái kết thúc.

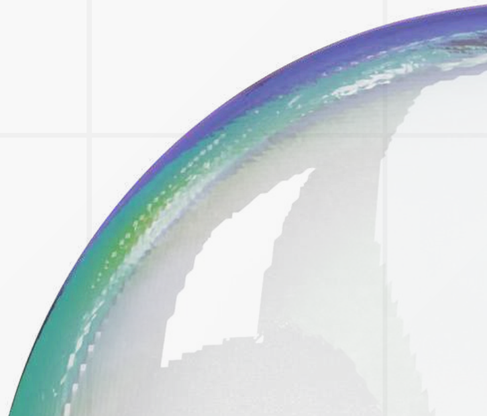
CÁC ĐIỂM QUAN TRỌNG

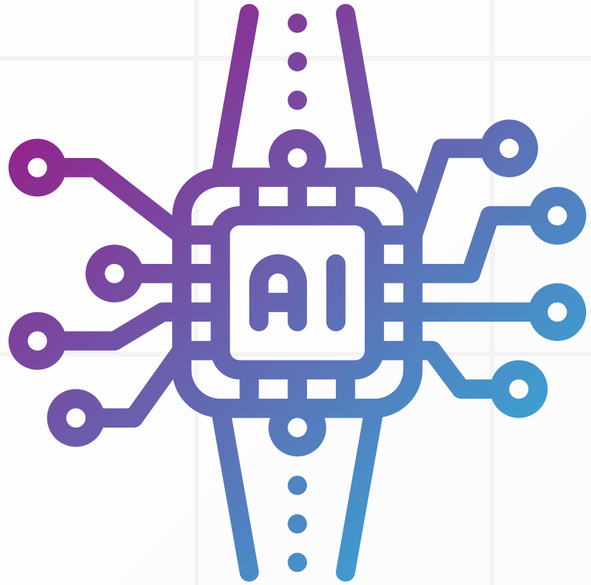
Phương pháp học củng cố (Reinforcement Learning Method)

Phần thưởng (Rewards)

Học hàm tiện ích và giá trị hành động (Learning Utility and Action Value Functions)

Khó khăn và ứng dụng của học củng cố (Challenges and Applications of Reinforcement Learning)



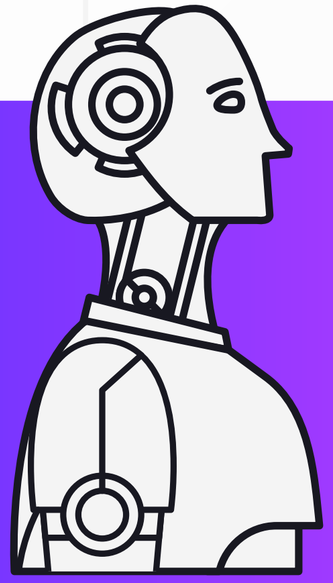


PASSIVE LEARNING IN A KNOWN ENVIRONMENT

Hệ thống nhận biết cố gắng học các tiện ích của các trạng thái và chuyển đổi trạng thái. Một mô hình xác suất chuyển đổi giữa các trạng thái được cung cấp.

Trong quá trình huấn luyện, hệ thống nhận biết di chuyển từ trạng thái ban đầu đến trạng thái kết thúc để nhận phần thưởng.

Mục tiêu là học các tiện ích kỳ vọng cho mỗi trạng thái không kết thúc, giả định rằng tiện ích của một chuỗi là tổng các phần thưởng tích lũy trong chuỗi đó.



Cập nhật ngây thơ (NAIVE UPDATING)

Naive Updating là phương pháp trong học tăng cường thụ động, tập trung vào việc cập nhật ước lượng tiện ích từ các phần thưởng quan sát được trong mỗi chuỗi huấn luyện.

Phương pháp này giảm thiểu sai số bằng cách duy trì trung bình chạy cho mỗi trạng thái và biến bài toán thành một bài toán học có giám sát.

Hạn chế

Biểu diễn mạnh mẽ hơn cho hàm tiện ích: Sử dụng mạng nơ-ron để học trực tiếp từ dữ liệu.

Phương pháp LMS và học củng cố: Bỏ qua mối quan hệ phụ thuộc giữa các tiện ích của các trạng thái.

Ràng buộc từ cấu trúc chuyển đổi giữa các trạng thái: Không xem xét ràng buộc này.

Hạn chế của phương pháp LMS: Dẫn đến hội tụ chậm trên các giá trị tiện ích chính xác.

PASSIVE LEARNING IN A KNOWN ENVIRONMENT

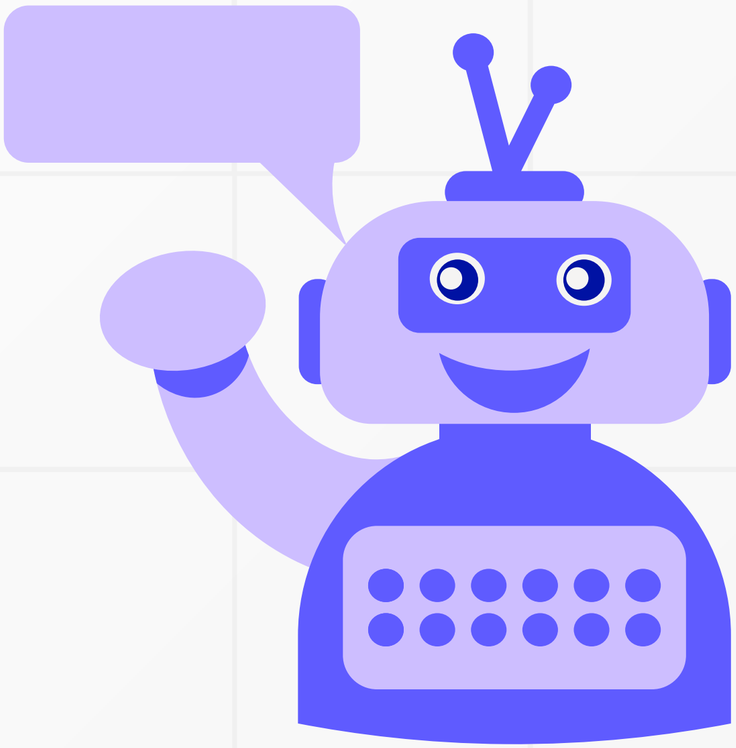
Adaptive dynamic programming
(Lập trình động thích ứng)

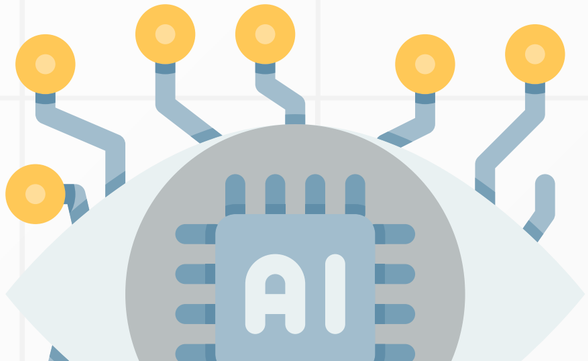
Nhanh hơn khi sử dụng kiến thức cấu trúc môi trường: Học nhanh hơn với kiến thức về xác suất chuyển đổi giữa các trạng thái.

Nhược điểm của LMS: Dẫn đến ước lượng tiện ích sai do bỏ qua mối quan hệ giữa các trạng thái.

Khắc phục nhược điểm: Sử dụng kiến thức về xác suất chuyển đổi để giải hệ phương trình tiện ích.

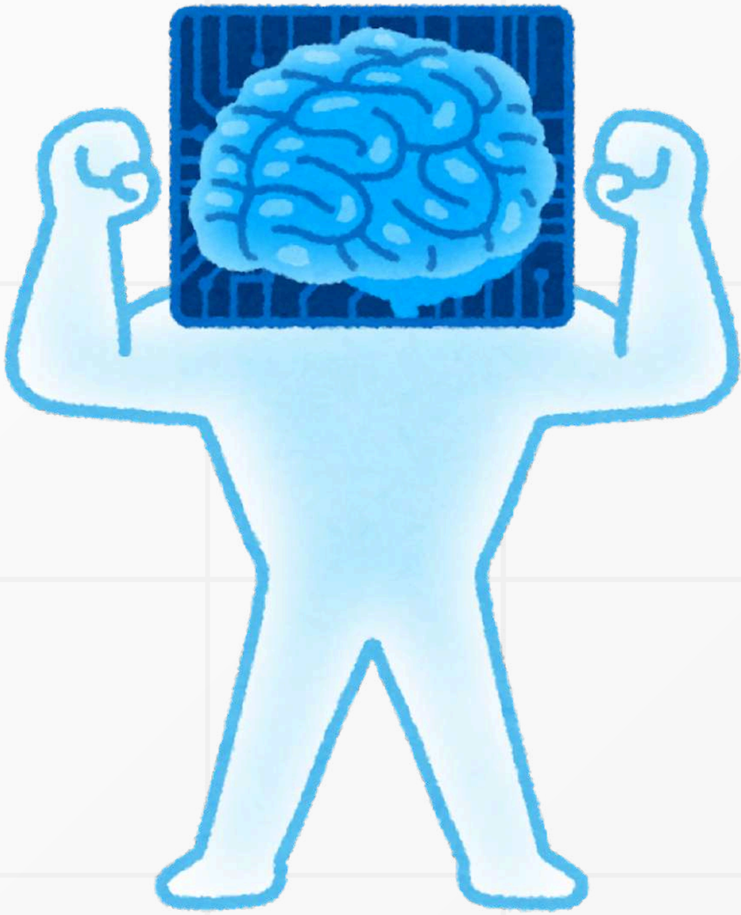
Lập trình động điều chỉnh (ADP): Phương pháp học củng cố giải phương trình tiện ích bằng thuật toán lập trình động, hiệu quả nhưng khó áp dụng cho không gian trạng thái lớn.





PASSIVE LEARNING IN A KNOWN ENVIRONMENT

Temporal difference learning
(Học tập khác biệt theo thời gian)



Xấp xỉ phương trình ràng buộc: Xấp xỉ các phương trình ràng buộc mà không cần giải chúng cho tất cả các trạng thái.

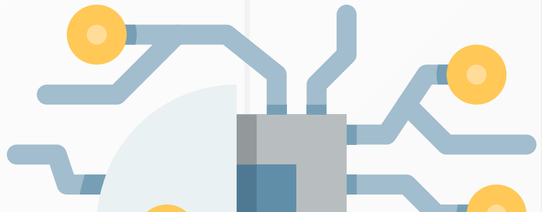
Quy tắc cập nhật TD (Temporal Difference): Sử dụng các chuyển đổi quan sát để điều chỉnh giá trị trạng thái dựa trên quy tắc cập nhật TD.

Ý tưởng của TD: Xác định điều kiện cân bằng và sử dụng phương trình cập nhật để điều chỉnh ước lượng tiện ích đến cân bằng lý tưởng.

Cập nhật trạng thái kế cận: Cập nhật chỉ liên quan đến trạng thái kế cận, nhưng giá trị trung bình hội tụ về giá trị chính xác.

Thay đổi α: Giảm dần α khi số lần ghé thăm trạng thái tăng, giúp hội tụ chính xác.

Thuật toán TD-UPDATE: Sử dụng quy tắc cập nhật TD để điều chỉnh tiện ích, có sai số RMS nhỏ hơn LMS sau số vòng lặp.



PASSIVE LEARNING IN A KNOWN ENVIRONMENT

PASSIVE LEARNING IN AN UNKNOWN ENVIRONMENT

Là quá trình học trong một môi trường không biết trước, nơi hệ thống nhận biết cố gắng học các tiện ích của các trạng thái mà không có thông tin trước về cấu trúc của môi trường.

Các phương pháp chính

Lập trình động (Dynamic Programming - DP)

Học theo sự khác biệt thời gian (Temporal-Difference - TD)

Phương pháp tối thiểu bình phương (Least Mean Squares - LMS)

Phương pháp ADP (Adaptive Dynamic Programming)

Lập trình động (Dynamic Programming - DP)

- Mô tả: Yêu cầu mô hình đầy đủ của môi trường.
- Ưu điểm:

  Tính toán chính xác, hội tụ nhanh với mô hình chính xác.
- Nhược điểm:

  Cần mô hình môi trường đầy đủ.

  Tốn kém tính toán, khó áp dụng cho không gian trạng thái lớn.

Học theo sự khác biệt thời gian (Temporal-Difference - TD)

- Mô tả: Dựa trên chuỗi quan sát để điều chỉnh ước lượng tiện ích.
- Ưu điểm:

  Không cần mô hình môi trường đầy đủ.

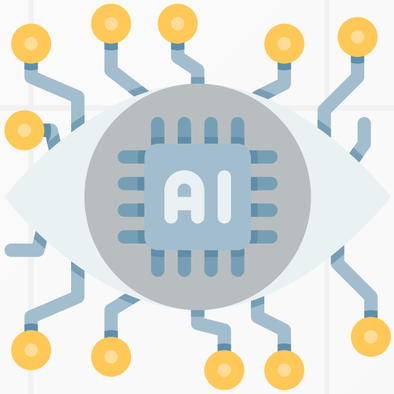
  Dựa trên trải nghiệm trực tiếp.
- Nhược điểm:

  Hội tụ chậm hơn DP.

  Có thể mắc kẹt nếu không đủ dữ liệu.

PASSIVE LEARNING IN AN UNKNOWN ENVIRONMENT

Phương pháp tối thiểu bình phương (Least Mean Squares - LMS)ng - DP)



- Mô tả: Giảm thiểu lỗi bình phương giữa giá trị ước lượng và giá trị thực tế.
- Ưu điểm:

  Dễ triển khai, đơn giản.

  Tốt cho môi trường liên tục và phi tuyến tính.
- Nhược điểm:

  Cần nhiều dữ liệu để hội tụ chính xác.

  Vấn đề với các mẫu không đại diện.

Phương pháp ADP (Adaptive Dynamic Programming)

- Mô tả: Kết hợp mô hình hóa môi trường và học từ trải nghiệm.
- Ưu điểm:

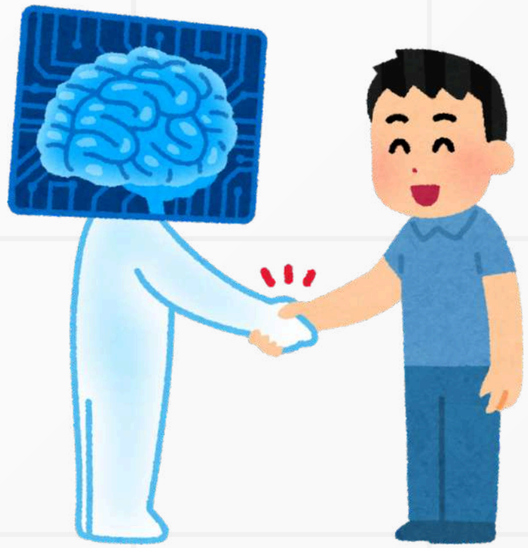
  Kết hợp ưu điểm của DP và TD.

  Hội tụ nhanh hơn TD.
- Nhược điểm:

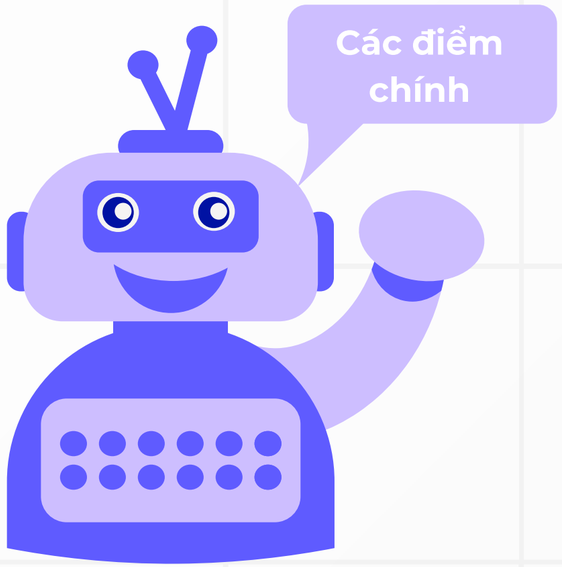
  Cần tính toán bổ sung để cập nhật mô hình môi trường.

PASSIVE LEARNING IN AN UNKNOWN ENVIRONMENT

ACTIVE LEARNING IN AN UNKNOWN ENVIRONMENT



Là quá trình mà một hệ thống nhận biết học cách điều chỉnh hành động dựa trên việc khám phá và học hỏi từ môi trường chưa biết.

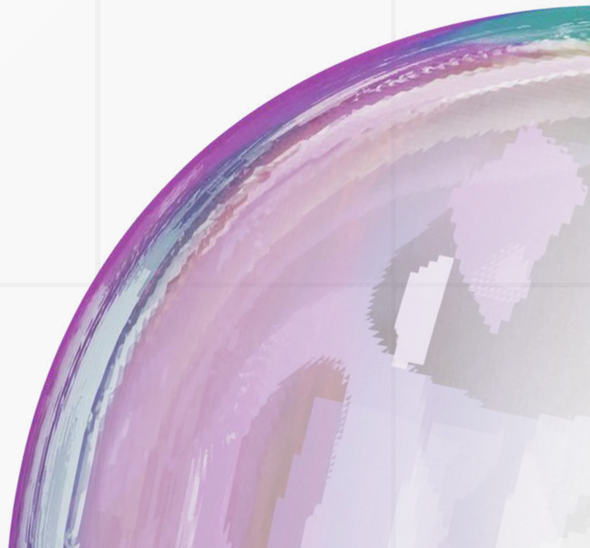


Mô hình môi trường: Tích hợp xác suất chuyển đổi dựa trên hành động cụ thể.

Ràng buộc tiện ích: Sử dụng phương trình phù hợp để tính tiện ích khi có sự lựa chọn hành động.

Hệ thống nhận biết học cộng cố: Học mô hình môi trường và tiện ích, điều chỉnh dựa trên hành động thực hiện, cập nhật mô hình và tính toán lại hàm tiện ích.

Hệ thống nhận biết học tiện ích thời gian rời rạc: Học mô hình để sử dụng hàm tiện ích trong quyết định, áp dụng quy tắc cập nhật và xử lý các kết quả không thường xuyên.



EXPLORATION

Là quá trình cân bằng giữa khám phá môi trường (kỳ dị) và khai thác phần thưởng (tham lam).

Kỳ dị: Khám phá ngẫu nhiên, dẫn đến hiệu suất học tốt hơn nhưng không tối ưu.

Tham lam: Tối đa hóa phần thưởng ngay lập tức, bỏ lỡ cơ hội học trong tương lai.

Phương pháp tốt nhất cân bằng giữa kỳ dị và tham lam, khám phá nhiều khi chưa biết môi trường và khai thác khi có mô hình gần đúng.

EXPLORATION AND BANDITS

- Exploration: Cân bằng giữa khám phá ngẫu nhiên và khai thác phần thưởng hiện tại.
- Bandit Problem: Lựa chọn hành động tối ưu từ một tập n tùy chọn để tối đa hóa tổng lợi ích kỳ vọng.
- Mục tiêu: Tối đa hóa tổng lợi ích kỳ vọng trong suốt cuộc đời của tác nhân.
- Thực tế: Tối ưu hóa phức tạp, kết quả tốt nhất đạt được trong giới hạn kinh nghiệm.

LEARNING AN ACTION-VALUE FUNCTION

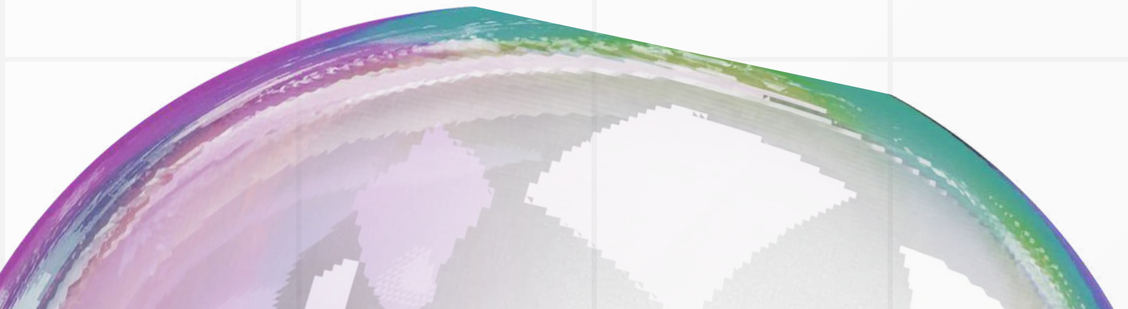
Q-learning là phương pháp linh hoạt, học giá trị Q mà không cần kiến thức trước về mô hình môi trường.

Giá trị Q: Kỳ vọng phần thưởng cho một hành động trong trạng thái, ký hiệu Q(a, i).

Liên hệ tiện ích: U(i) = max_a Q(a, i).

Tầm quan trọng: Quyết định không cần mô hình, học từ phản hồi phần thưởng.

Phương trình ràng buộc: Giá trị Q phải thỏa mãn tại cân bằng.



Q-learning: Cập nhật chênh lệch thời gian để tính Q, không cần mô hình.

Hiệu suất: Mô hình học và hàm tiện ích có thể hiệu quả hơn học hàm giá trị hành động không có mô hình.

Khám phá Q-learning: Sử dụng hàm thăm dò, không cần học mô hình chuyển đổi.

GENERALIZATION IN REINFORCEMENT LEARNING

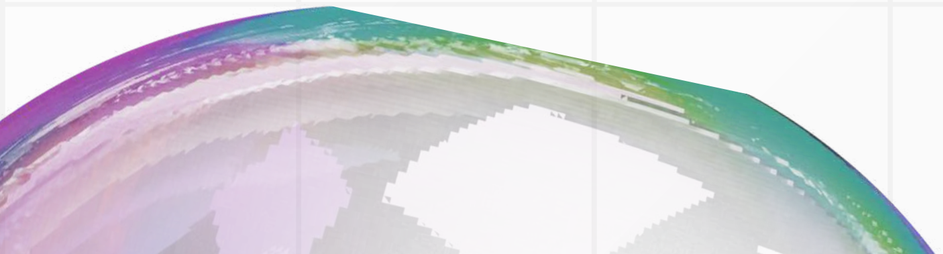
Biểu diễn ngầm: Sử dụng biểu diễn ngầm cho hàm tiện ích và giá trị hành động để giảm chi phí tính toán và tăng khả năng tổng quát hóa.

Biểu diễn tuyến tính: Hàm tiện ích có thể được biểu diễn như một hàm tuyến tính của các đặc trưng của môi trường.

Phương trình cập nhật TD: Sử dụng quy tắc cập nhật gradient descent trong không gian trọng số để giảm thiểu sai lệch cục bộ trong ước tính tiện ích.

Applications to game-playing

Application to robot control



Khám phá Q-learning: Sử dụng hàm thăm dò, không cần học mô hình chuyển đổi.

GENETIC ALGORITHMS AND EVOLUTIONARY PROGRAMMING

Định nghĩa từ tiến hóa tự nhiên: Sử dụng nguyên tắc lựa chọn tự nhiên được mô tả trong sách "The Origin of Species" của Darwin để phát triển thuật toán genetic algorithm.

Quy trình tiến hóa: Bắt đầu với một tập hợp các cá thể và sử dụng các toán tử lựa chọn và sinh sản để tiến hóa các cá thể dựa trên một hàm thích nghi.

Biểu diễn cá thể: Mỗi cá thể được biểu diễn dưới dạng một chuỗi gen trên một bảng chữ cái hữu hạn, thường là bảng chữ cái nhị phân.

Lựa chọn và sinh sản: Lựa chọn cá thể thường là ngẫu nhiên, với xác suất tỉ lệ thuận với thích nghi của mỗi cá thể, và sau đó thực hiện sinh sản thông qua sự giao phối và đột biến.

Hiệu suất đa dạng: Hiệu suất của thuật toán có thể thay đổi đáng kể trên các vấn đề cụ thể, do đó cần thử nghiệm trước khi đầu tư quá nhiều thời gian và công sức.

2.KNOWLEDGE IN LEARNING

INTRODUCTION

Đặc trưng logic và học cảm ứng: Đặc trưng logic của vấn đề học tập cho phép chỉ định thông tin về hàm cần học.

Mục tiêu của học cảm ứng: Tìm giả thuyết giải thích phân loại dựa trên các mô tả, được biểu diễn logic bằng công thức "Hypothesis A Descriptions |= Classifications"

Some Simple Examples

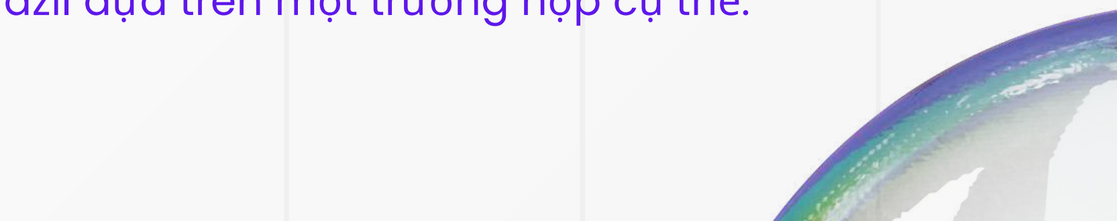
Nướng Thằn Lằn: nướng thằn lằn bằng đầu cây, dẫn đến kết luận rằng bất kỳ vật thể nào cũng có thể dùng để nướng thực phẩm mềm nhỏ.

Du Khách Ở Brazil: Một du khách kết luận rằng người Brazil nói tiếng Bồ Đào Nha dựa trên rằng mọi người trong một quốc gia thường nói cùng một ngôn ngữ.

Some general schemes

Nướng Thằn Lằn: Học dựa trên giải thích (EBL) biến nguyên tắc đầu tiên thành kiến thức cụ thể.

Du Khách Ở Brazil: Học dựa trên sự liên quan (RBL) cho phép du khách kết luận về ngôn ngữ nói chung của người Brazil dựa trên một trường hợp cụ thể.



EXPLANATION-BASED LEARN

Học dựa trên giải thích (EBL) trích xuất quy tắc tổng quát từ các quan sát cá nhân, giúp giải quyết vấn đề phức tạp bằng cách tổng quát hóa và tái sử dụng các nguyên tắc đã hiểu.

Extracting general rules from examples

Học dựa trên giải thích (EBL) xây dựng quy tắc tổng quát từ các chứng minh cụ thể, loại bỏ điều kiện không cần thiết để tạo ra quy tắc hiệu quả hơn.

Quy trình cơ bản của EBL

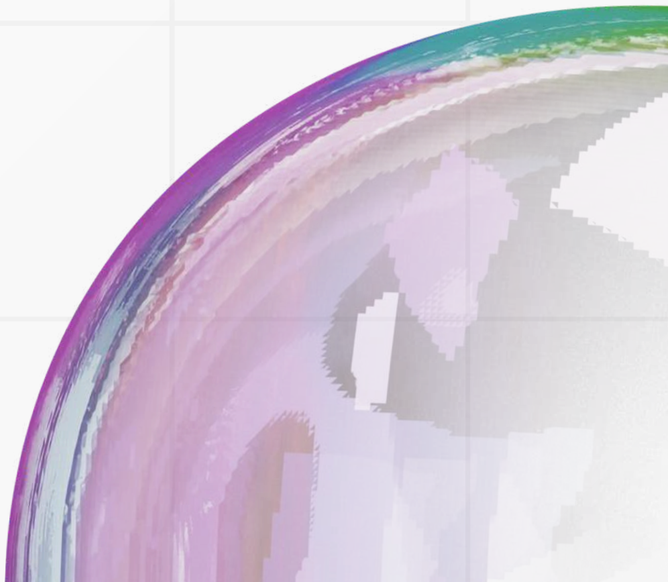
Bao gồm xây dựng chứng minh cho ví dụ cụ thể, sau đó tổng quát hóa để tạo ra một quy tắc mới, loại bỏ các điều kiện đúng, và sử dụng lại quy tắc này cho các tình huống tương tự.

Improving efficiency

Để cải thiện hiệu suất của EBL, cần cân nhắc giữa tính khả thi và tính tổng quát của các quy tắc trích xuất, tránh việc thêm quá nhiều quy tắc có thể làm chậm quá trình suy luận.

Một cách tiếp cận phổ biến là đảm bảo tính khả thi của mỗi mục tiêu phụ trong quy tắc, mặc dù cần xem xét độ phức tạp và chi phí giải quyết mỗi mục tiêu phụ.

Phân tích thực nghiệm về hiệu suất là trung tâm của EBL, giúp tạo ra một cơ sở tri thức hiệu quả hơn cho các vấn đề trong tương lai.



EXTRACTING GENERAL RULES FROM EXAMPLES

LEARNING USING RELEVANCE INFORMAT

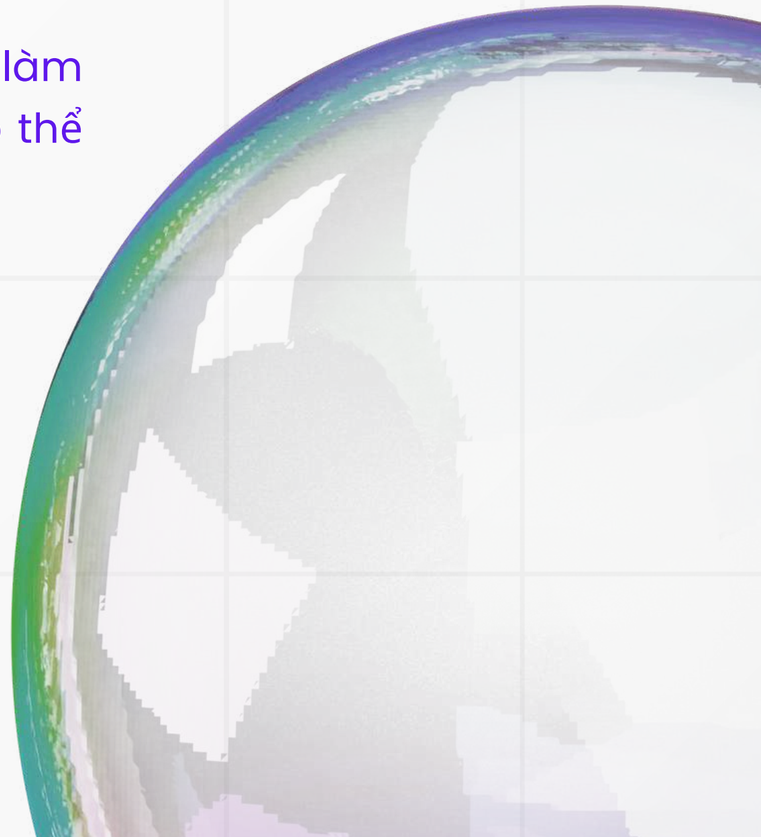
Learning Using Relevance Information (LURI) là một phương pháp học máy sử dụng thông tin về mức độ liên quan của các đặc điểm (features) để cải thiện quá trình học.

Determining the hypothesis space

Xác định chức năng giữa các thuộc tính giúp hạn chế không gian các giả thuyết, làm cho việc học dễ dàng hơn. Sử dụng kết quả cơ bản của lý thuyết học máy, ta có thể định lượng lợi ích của việc này.

Learning and using relevance information

Thuật toán học được trình bày trong phần này tìm cách tìm ra xác định đơn giản nhất phù hợp với các quan sát.



INDUCTIVE LOGIC PROGRAMMING

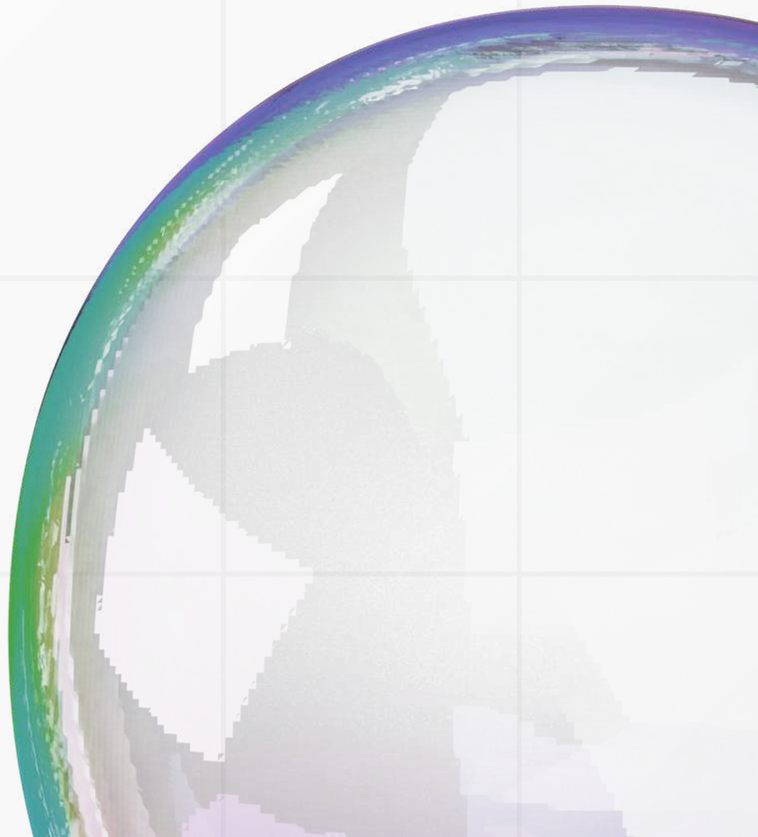
Lập trình logic trong trí tuệ nhân tạo (ILP) kết hợp phương pháp thuận dụng và biểu diễn bậc nhất, tạo ra các chương trình logic để giải quyết vấn đề tổng quát trong học máy và suy luận từ dữ liệu.

Là phương pháp nghiêm ngặt và hoàn chỉnh cho học từ dữ liệu, đặc biệt hiệu quả trong các lĩnh vực mà các phương pháp khác gặp khó khăn.

Inverse resolution

Ngược định lý trong học dựa trên logic cho phép tìm ra giả thuyết phù hợp bằng cách chạy ngược lại quá trình giải quyết.

- **Generating inverse proofs**
- **Discovering new predicates and new knowledge**



Generating inverse proofs

Là phương pháp tách mệnh đề hợp thành các mệnh đề hoặc ngược lại, là một quá trình tìm kiếm đòi hỏi lựa chọn mỗi bước, thường được cải thiện bằng cách áp dụng các hạn chế và chiến lược giải quyết.

Discovering new predicates and new knowledge

Dựa trên logic có thể tạo ra các giả thuyết từ các ví dụ bằng cách đảo ngược các chiến lược giải quyết, thậm chí cung cấp khả năng phát minh các predicate mới để giải thích dữ liệu.

Tuy nhiên, áp dụng chúng vào các nhiệm vụ phức tạp đòi hỏi sự cải tiến liên tục.

INDUCTIVE LOGIC PROGRAMMING

Top-down learning methods

Phương pháp học từ trên xuống trong ILP tập trung vào việc tạo ra các quy tắc tổng quát từ một quy tắc khởi đầu và điều chỉnh chúng để phù hợp với dữ liệu.

FOIL là một ví dụ điển hình, xây dựng các mệnh đề từng bước một và sử dụng một phép chọn lọc để tạo ra các quy tắc phân loại đúng cho các ví dụ.

Thank you!