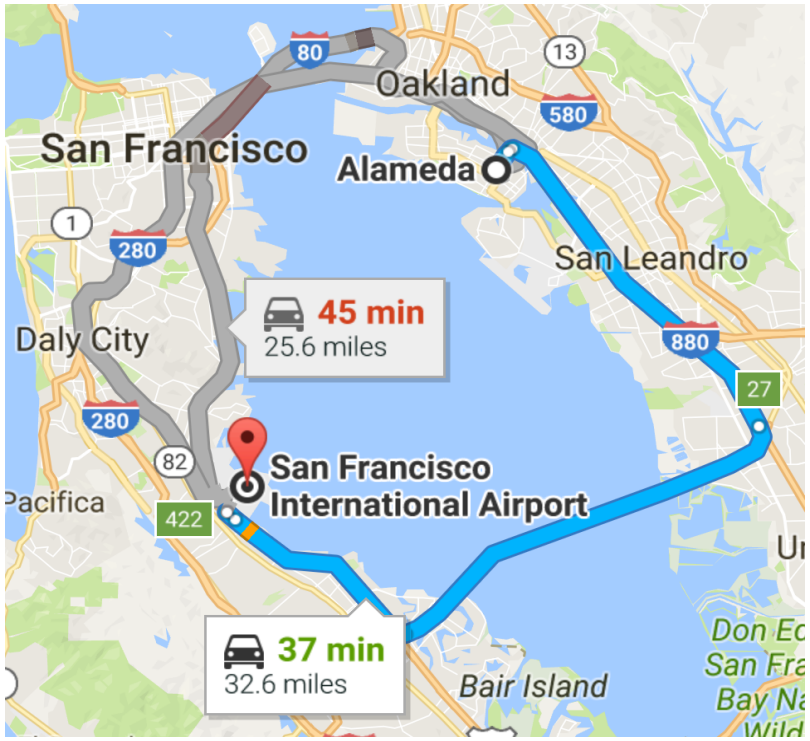
On Learning Sparse Boolean Formulae For Explaining AI Decisions

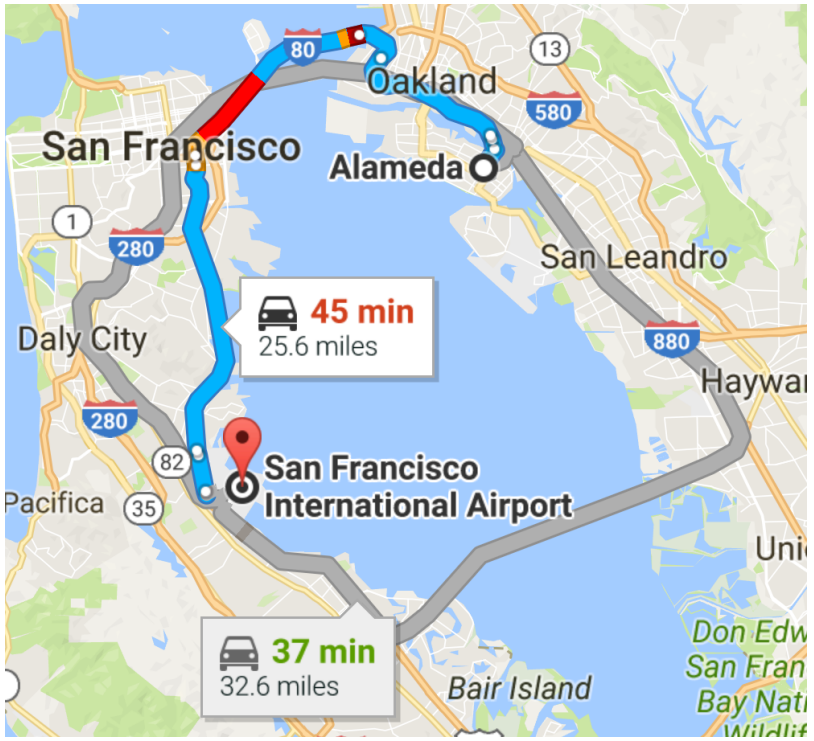
SUSMIT JHA, SRI INTERNATIONAL

VASUMATHI RAMAN, ZOOX INC.

ALESSANDRO PINTO, TUHIN SAHAI, AND MICHAEL FRANCIS
UNITED TECHNOLOGIES RESEARCH CENTER, BERKELEY

Ubiquitous AI and need for explaination



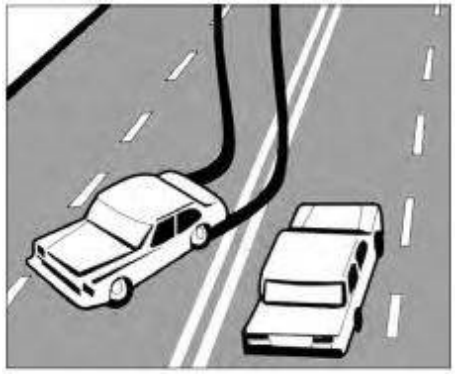


- This route is faster.
- There is traffic on Bay Bridge.
- There is an accident just after Bay Bridge backing up traffic.

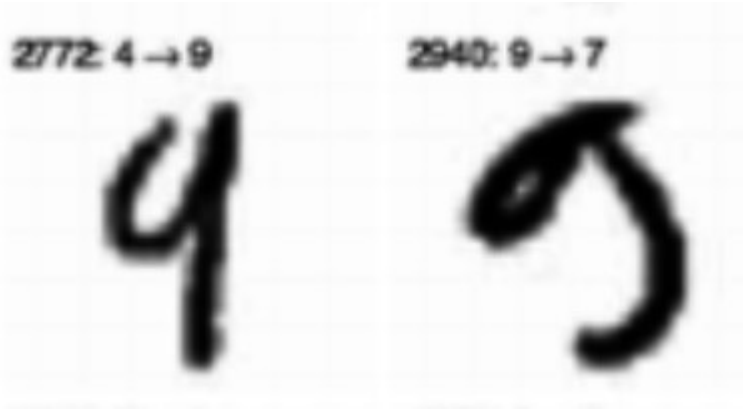
Why did we take the San Mateo bridge instead of the Bay Bridge ?

Decision/Recommendation/Classification

Medical Diagnosis



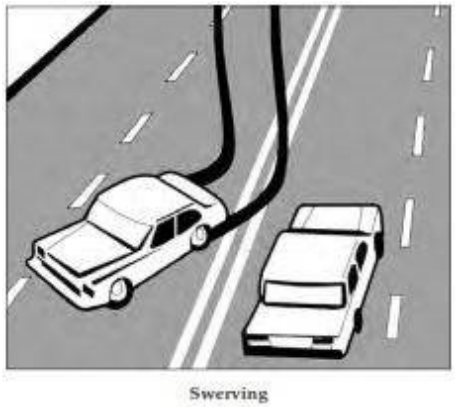
Autonomous Systems Certification



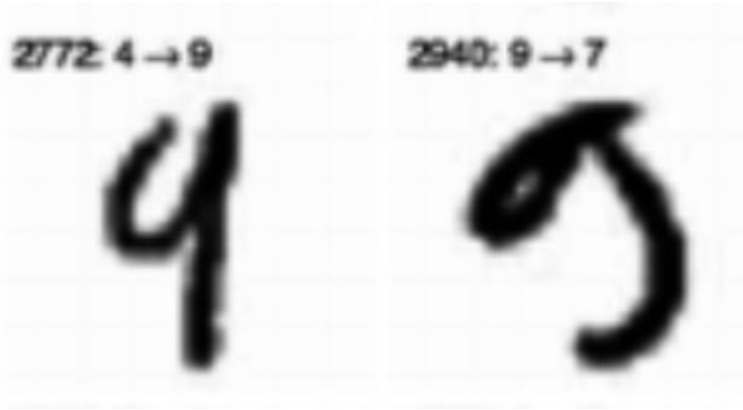
Incorrect ML Recommendations

Decision/Recommendation/Classification

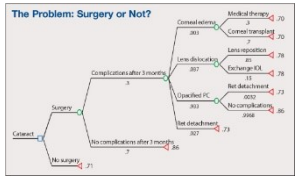
Medical Diagnosis

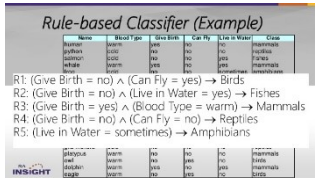


Autonomous Systems Certification

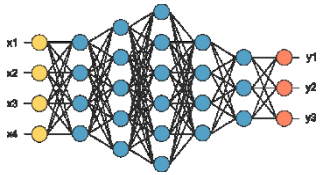


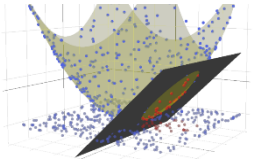
Incorrect ML Recommendations





Interpretable but less scalable:
Decision Trees, Propositional Rules





Scalable but less interpretable :
Neural Networks, Support Vector

Even `algorithmic' decision making: A* Path Planning

A* is an algorithm that:

Uses heuristic to guide search

While ensuring that it will compute a path with minimum cost

A* computes the function f(n) = g(n) + h(n)

"estimated cost"

"actual cost"

f(n) = g(n) + h(n)

g(n) = "cost from the starting node to reach n"

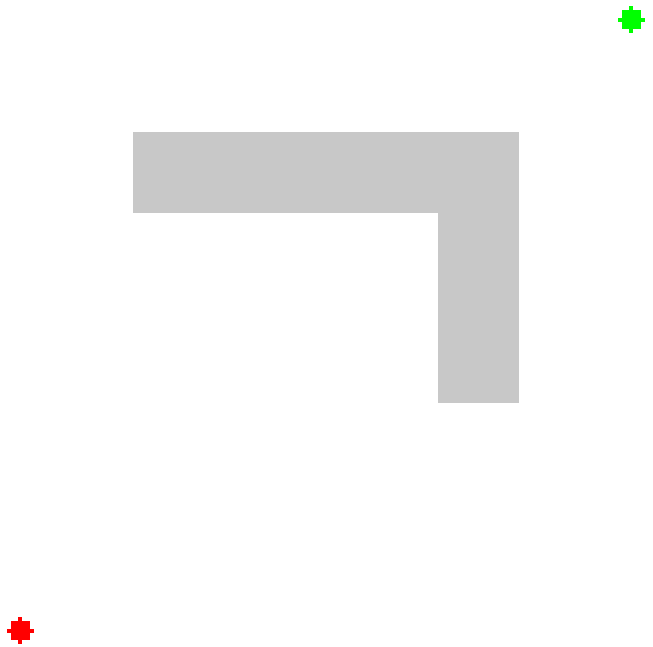
h(n) = "estimate of the cost of the cheapest path from n to the goal node"

Algorithm 1: A*

Input: start, $goal(n)$, $h(n)$, $expand(n)$
Output: path

```
1  if goal(start) = true then return makePath(start)
2
3  open ← start
4  closed ← ∅
5  while open ≠ ∅ do
6      sort(open)
7      n ← open.pop()
8      kids ← expand(n)
9      forall the kid ∈ kids do
10         kid.f ← (n.g + 1) + h(kid)
11         if goal(kid) = true then return makePath(kid)
12         if kid ∩ closed = ∅ then open ← kid
13     closed ← n
14 return ∅
```

Example: A* Path Planner

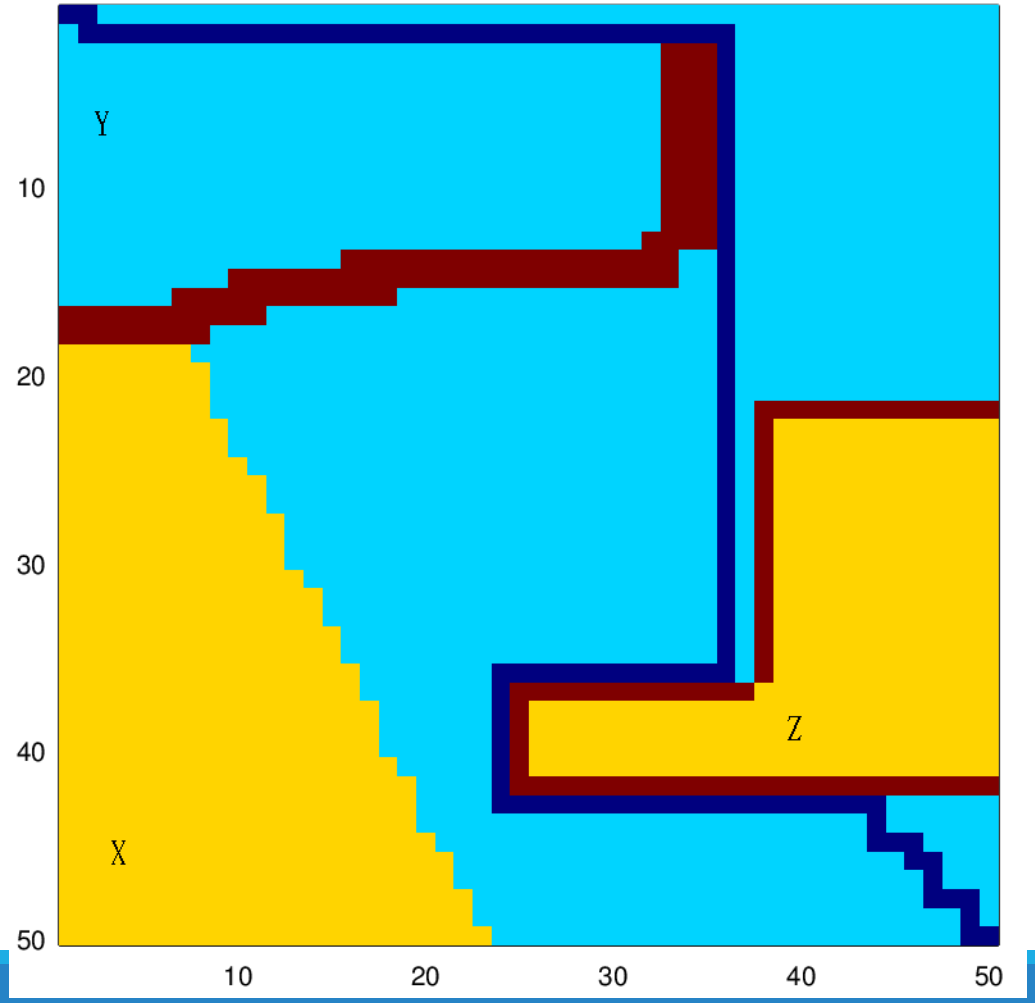


Algorithm 1: A*

Input: start, $goal(n)$, $h(n)$, $expand(n)$
Output: path

```
1  if goal(start) = true then return makePath(start)
2
3  open ← start
4  closed ← ∅
5  while open ≠ ∅ do
6      sort(open)
7      n ← open.pop()
8      kids ← expand(n)
9      forall the kid ∈ kids do
10         kid.f ← (n.g + 1) + h(kid)
11         if goal(kid) = true then return makePath(kid)
12         if kid ∩ closed = ∅ then open ← kid
13     closed ← n
14 return ∅
```

Example: A* Path Planner



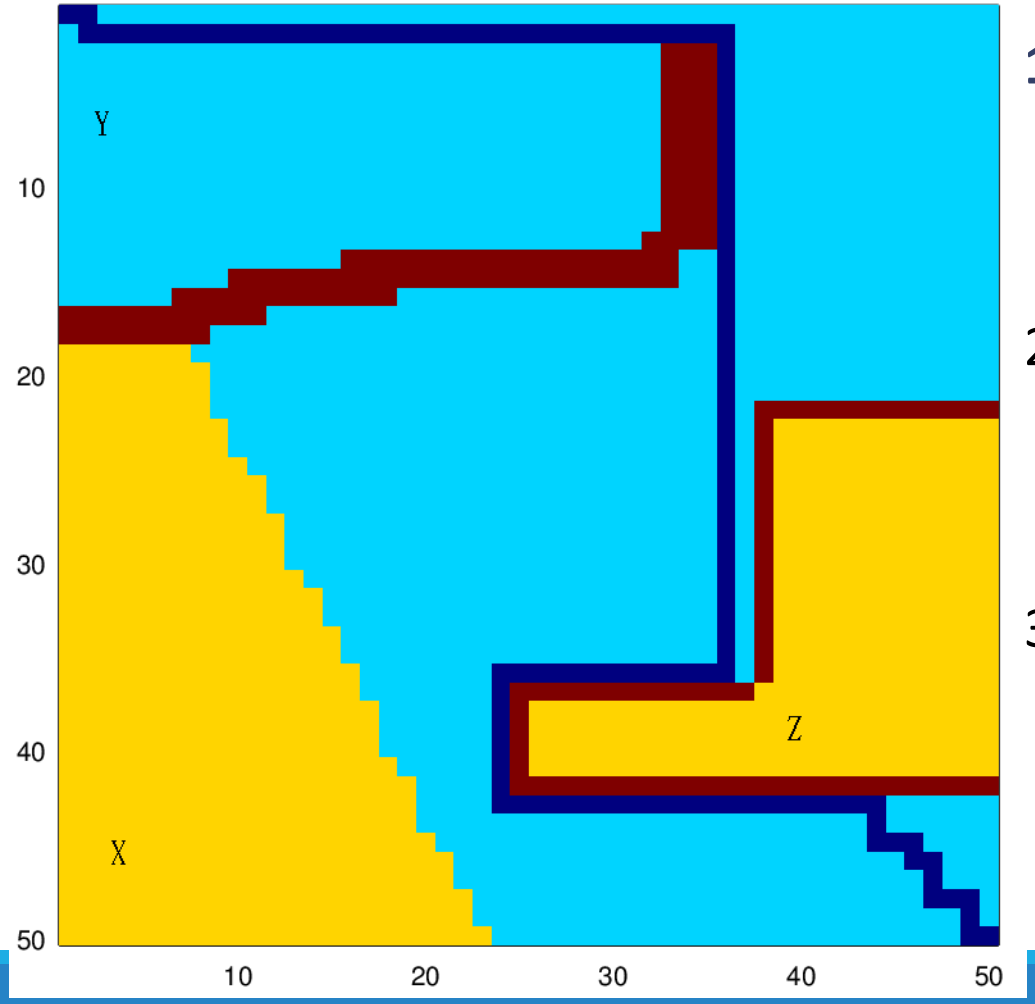
Algorithm 1: A*

Input: start, $goal(n)$, $h(n)$, $expand(n)$
Output: path

```
1  if goal(start) = true then return makePath(start)
2
3  open ← start
4  closed ← ∅
5  while open ≠ ∅ do
6      sort(open)
7      n ← open.pop()
8      kids ← expand(n)
9      forall the kid ∈ kids do
10         kid.f ← (n.g + 1) + h(kid)
11         if goal(kid) = true then return makePath(kid)
12         if kid ∩ closed = ∅ then open ← kid
13     closed ← n
14 return ∅
```

Why didn’t we go through X / Z ?

Example: A* Path Planner

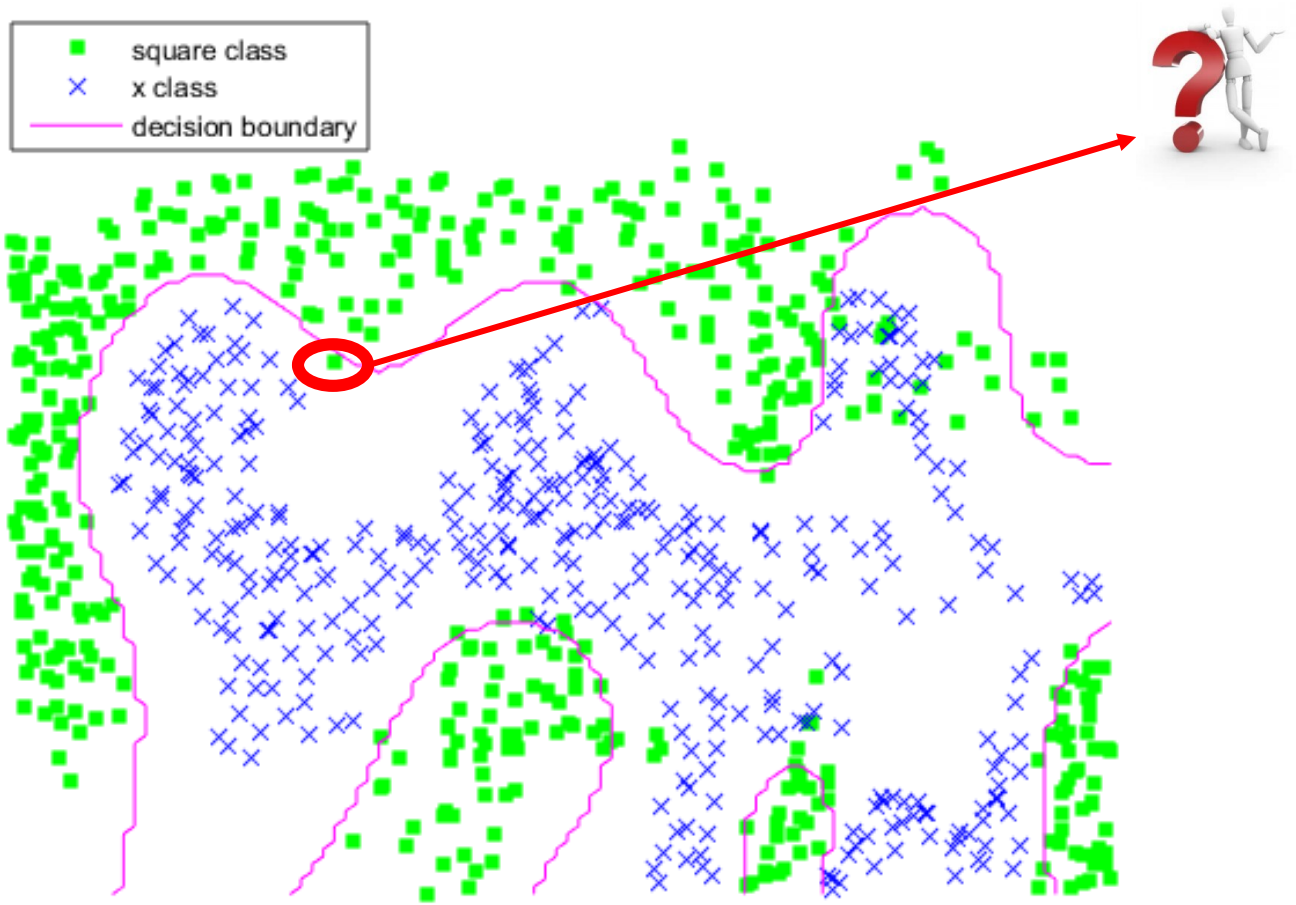


1. **Internal details** of algorithm and its implementation is unknown to human observer/user. So, explanation must be in terms of a **common vocabulary**.

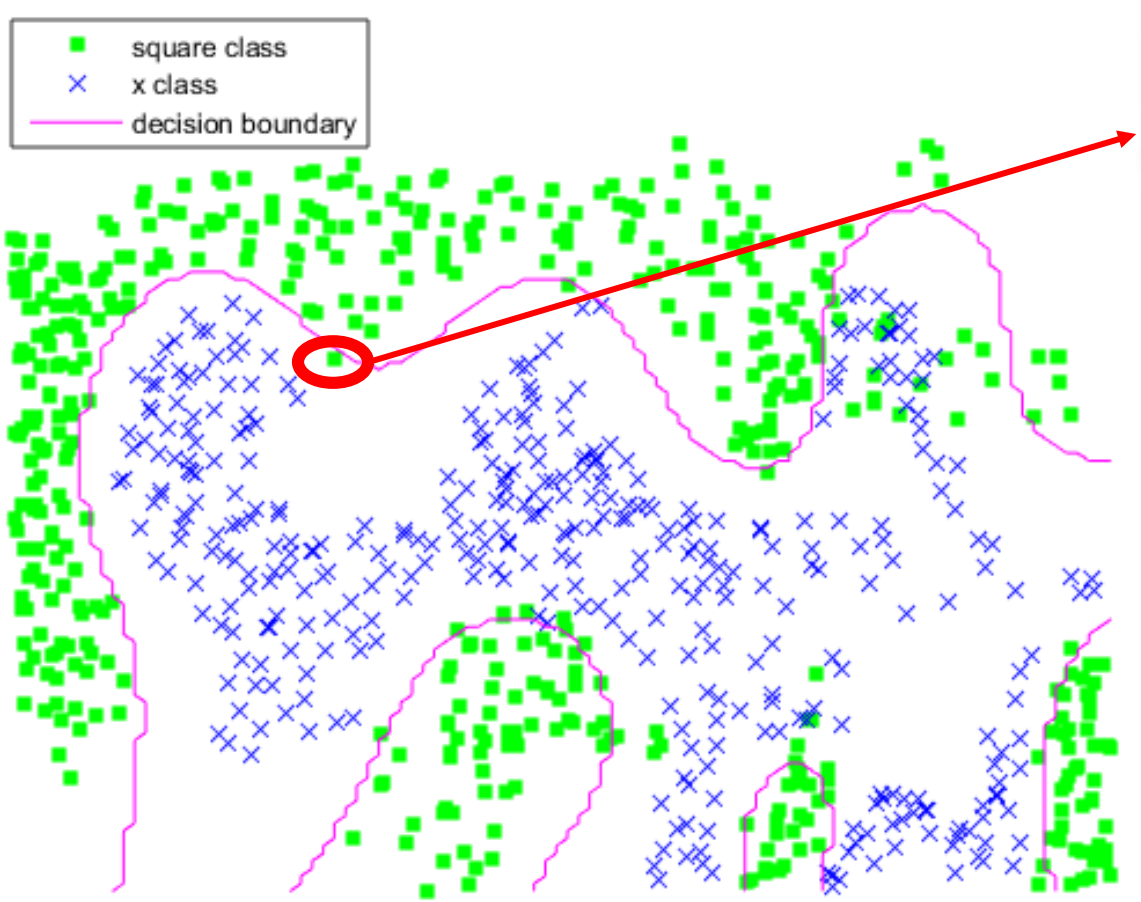
2. Some explanations need further **deduction and inference that were not performed** by the AI algorithm while making the original decision.

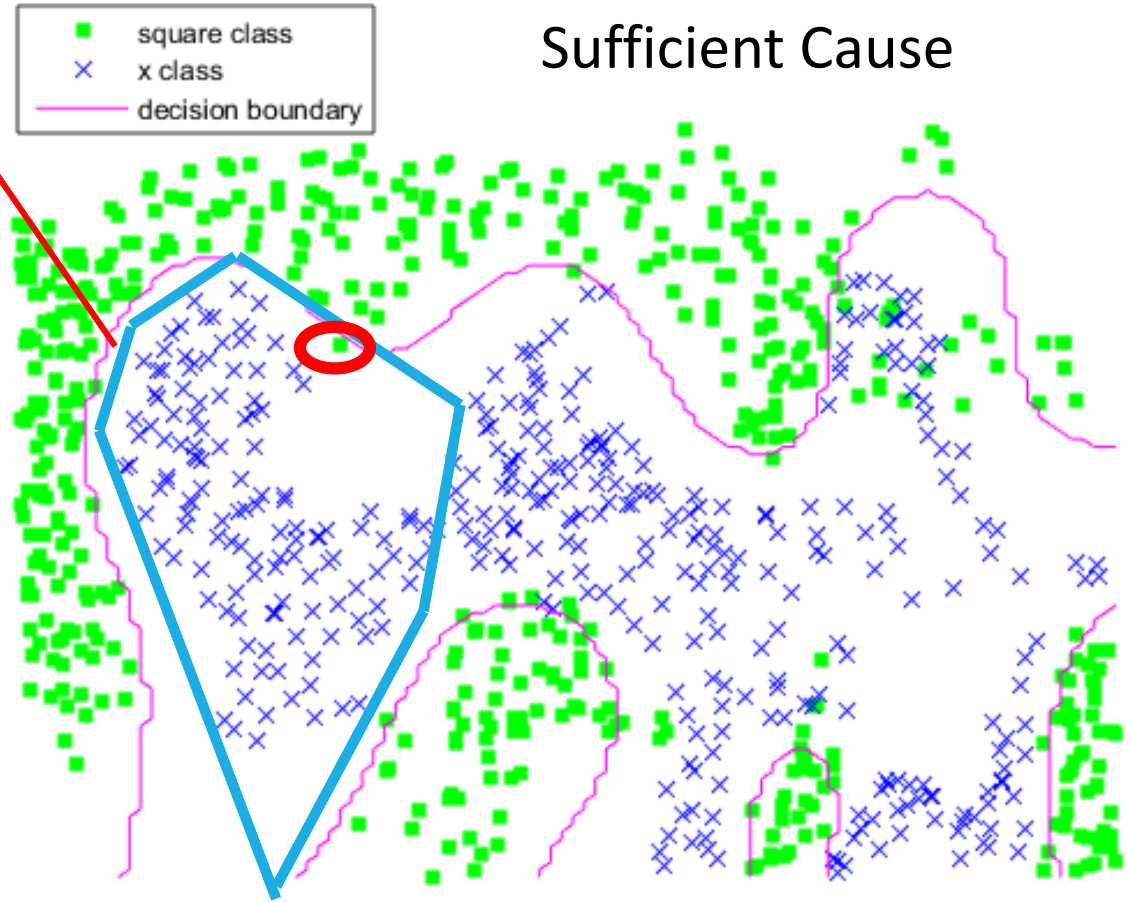
3. Decision making is often accomplished through complex composition of a number of AI algorithms and hence, an explanation process would be practical only if it **did not require detailed modeling of each AI algorithm**.

Local Explanations of Complex Models



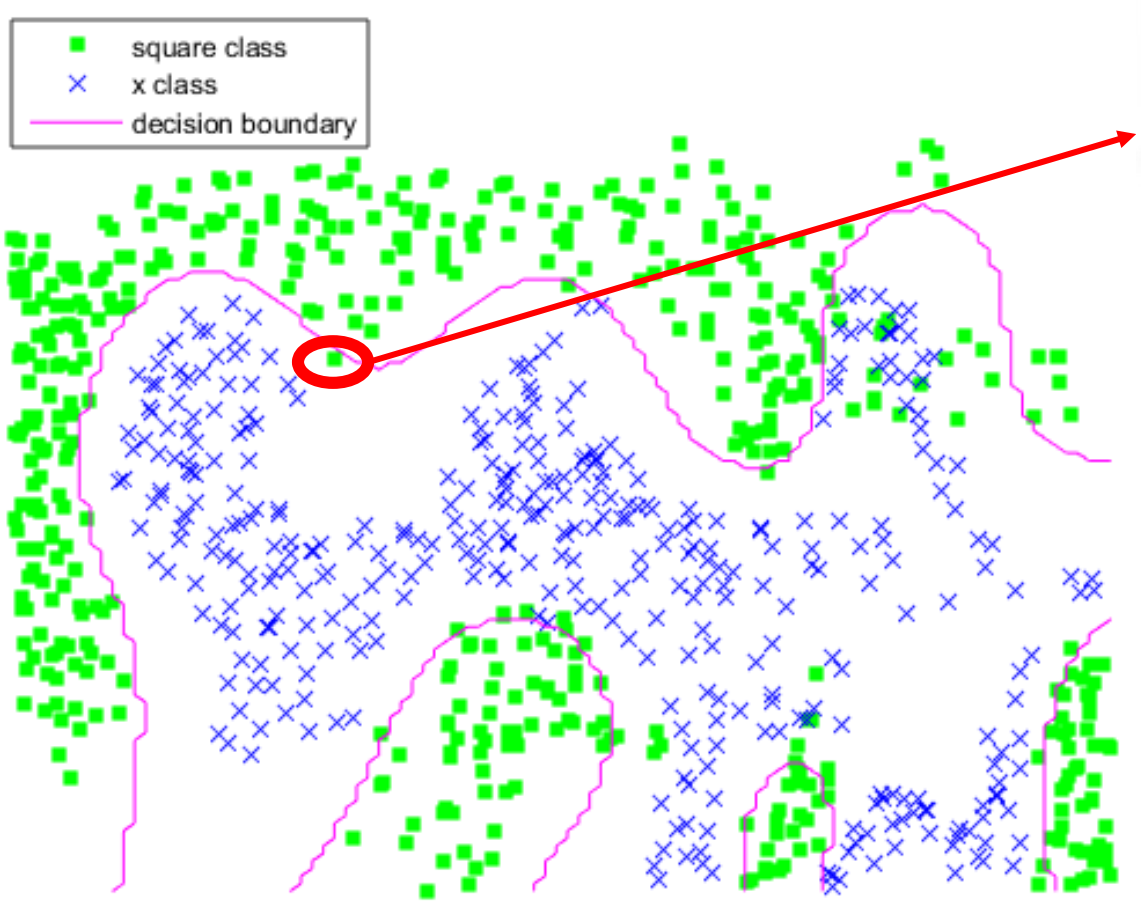
Local Explanations of Complex Models

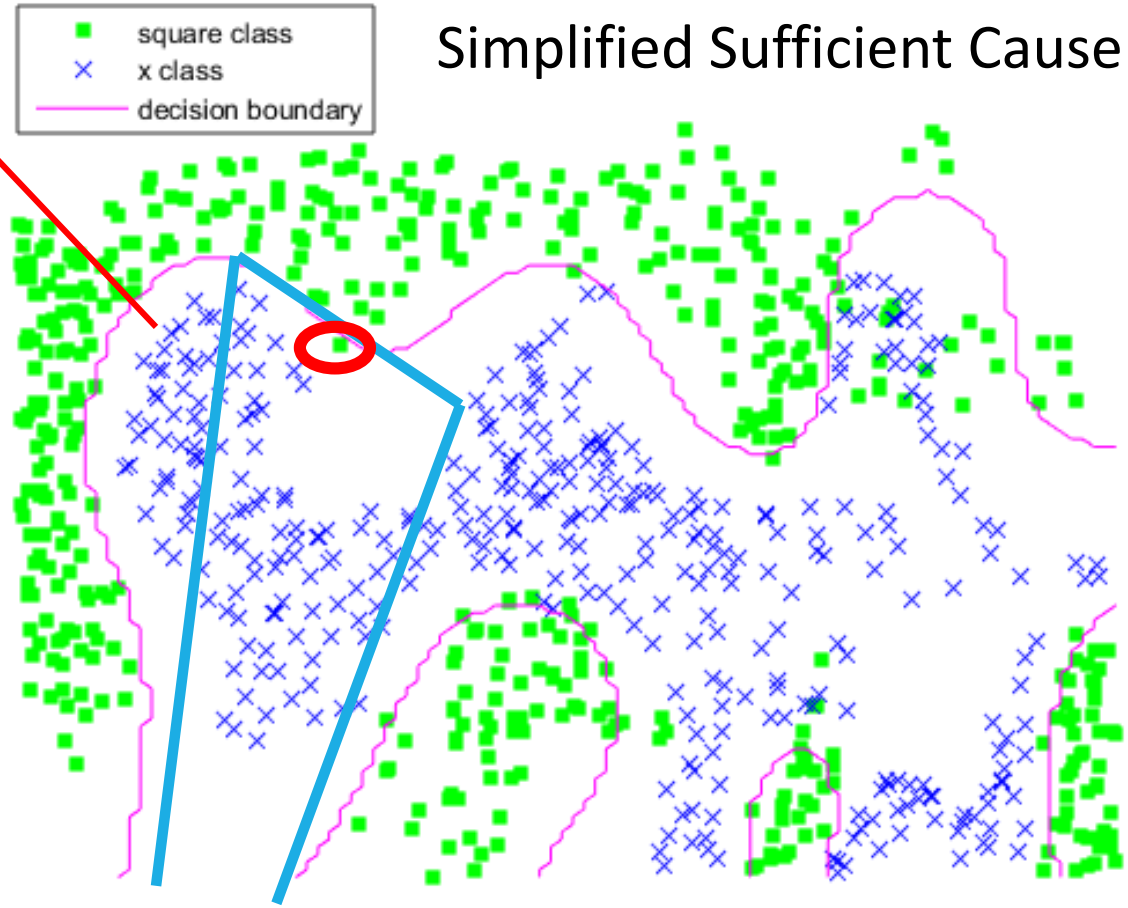




Sufficient Cause

Local Explanations of Complex Models





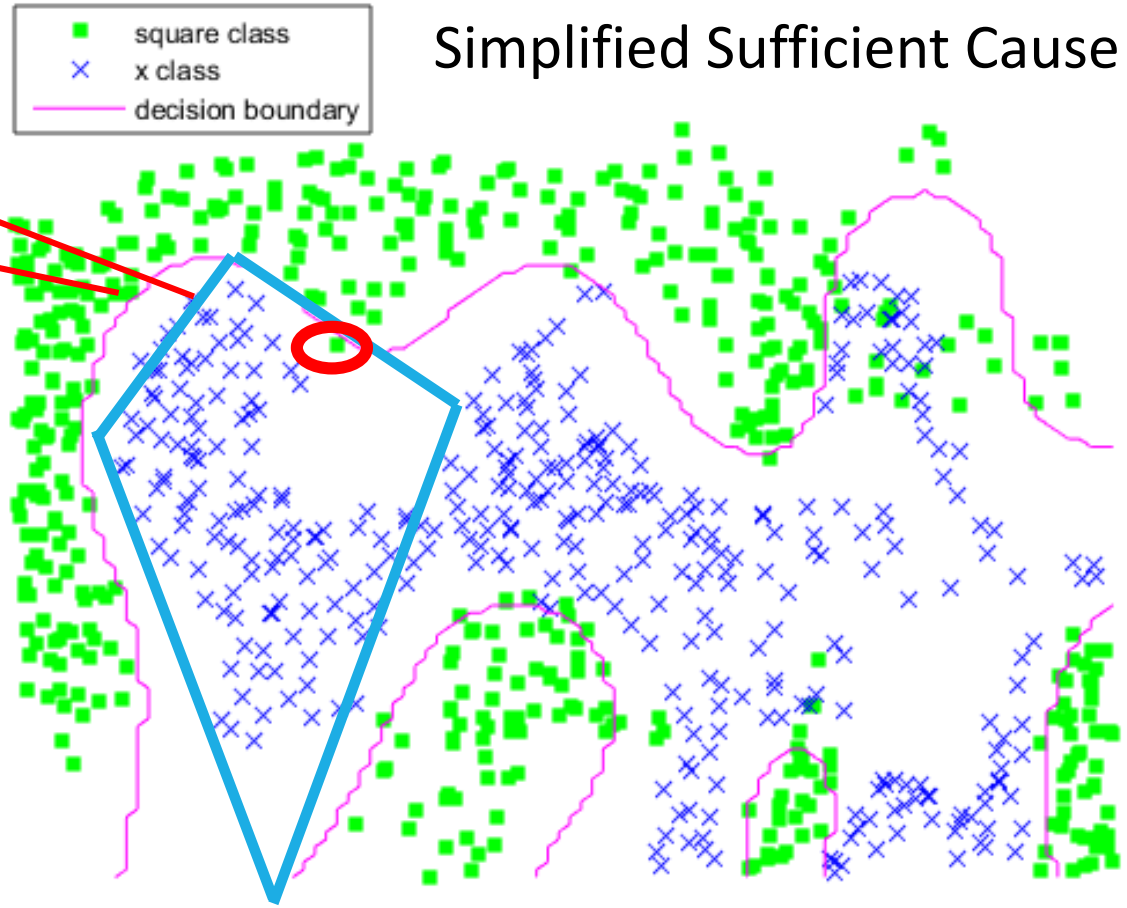
Simplified Sufficient Cause

Local Explanations in AI

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$\mathcal{L}(f, g, \pi_x)$ Measure of how well g approximates f

$\Omega(g)$ Measure of complexity of g

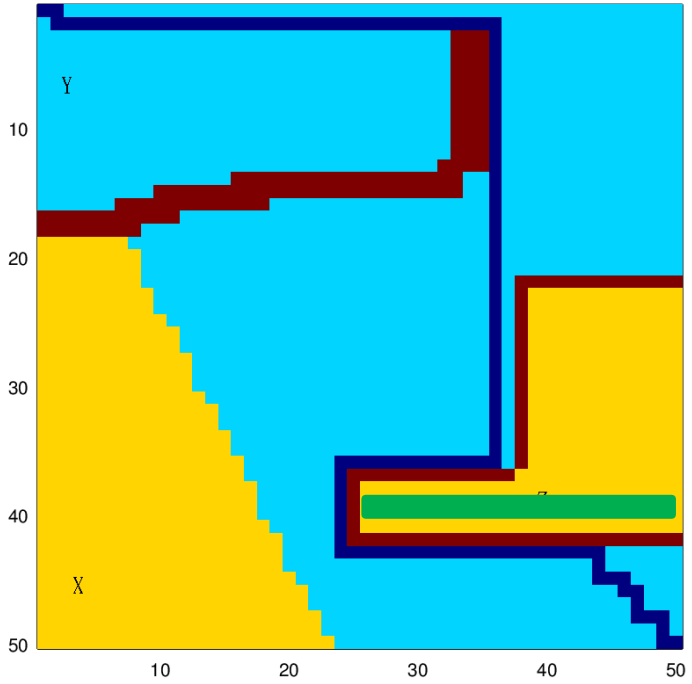


Simplified Sufficient Cause

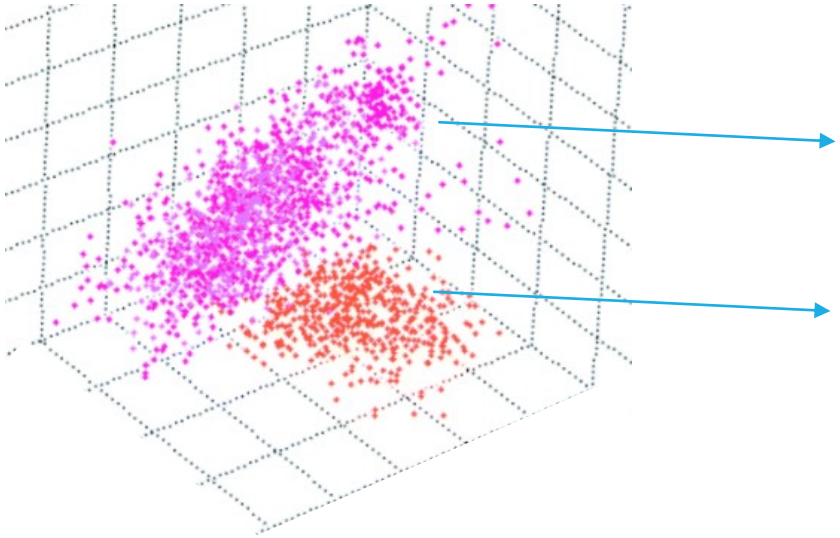
Formulation in AI:

- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?: Explaining the Predictions of Any Classifier." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
- Hayes, Bradley, and Julie A. Shah. "Improving Robot Controller Transparency Through Autonomous Policy Explanation." *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2017.

Model Agnostic Explanation through Boolean Learning



Why does the path not go through Green?



Let each point in k-dimensions (for some k) correspond to a map.

Maps in which optimum path goes via green

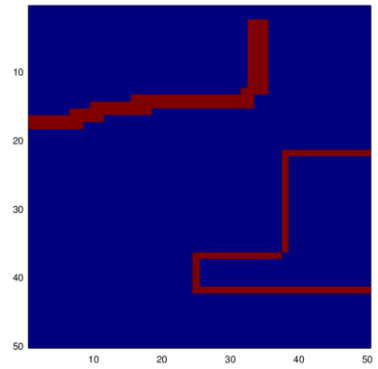
Maps in which optimum path does not go via green

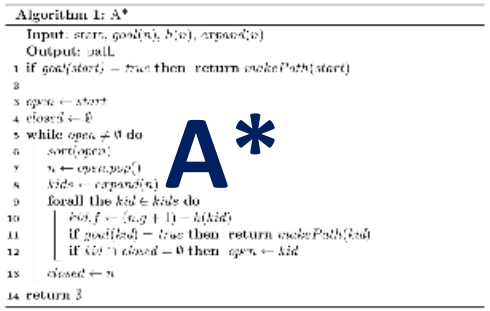
Find a Boolean formula ϕ such that

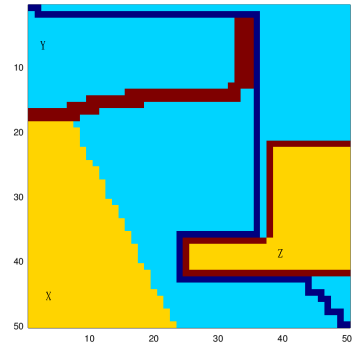
$\phi \Leftrightarrow Path\ contain\ z$

$\phi \Rightarrow Path\ contain\ z$

Example: Explanations in A*





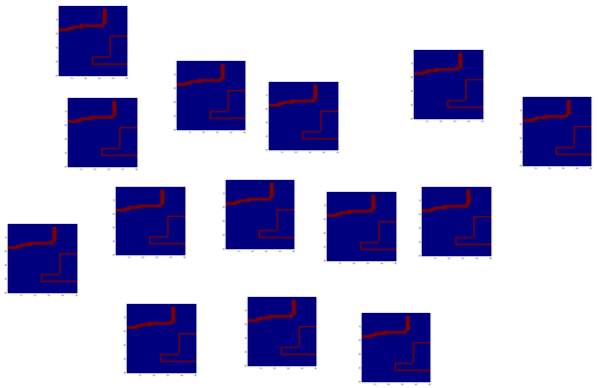


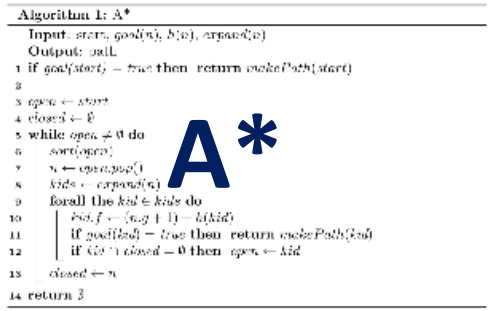
$\phi_{explain}$:
Using explanation vocabulary
Ex: Obstacle presence

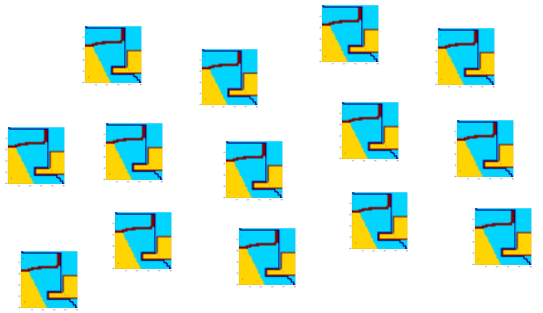
ϕ_{query} :
Some property of the output
Ex: Some cells not selected

$$\phi_{explain} \Rightarrow \phi_{query}$$
$$\phi_{explain} \Leftrightarrow \phi_{query}$$

Example: Explanations in A*







$\phi_{explain}$:
Using explanation vocabulary
Ex: Obstacle presence

ϕ_{query} :
Some property of the output
Ex: Some cells not selected

***Learn Decision Trees* For $\phi_{explain}$**
***Using Labels for* ϕ_{query}**

Example: Explanations in A*

50x50 grid has $2^{2^{50X50}}$ possible explanations even if vocabulary only considers presence/absence of obstacles.

Scalability: Usually the feature space or vocabulary is large. For a map, its order of features in the map. For an image, it is order of the image's resolution.

Guarantee: Is the sampled space of maps enough to generate the explanation with some quantifiable probabilistic guarantee?

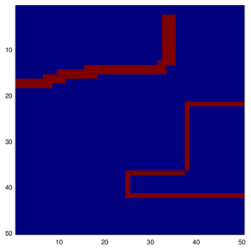
$\boldsymbol{Learn\ Decision\ Trees}$ **For** $\boldsymbol{\phi_{explain}}$
$\boldsymbol{Using\ Labels\ for\ \phi_{query}}$

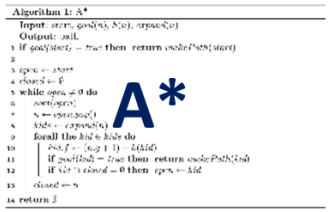
Example: Explanations in A*

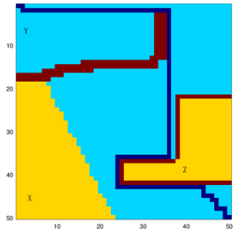
Definition *(PAC learnability) Let $\mathcal{C}$ be a class of boolean functions $f : \{0,1\}^n \rightarrow 0,1$. We say that $\mathcal{C}$ is PAC-learnable if there exists an algorithm $\mathcal{L}$ such that*
for every $f \in \mathcal{C}$
for any probability distribution $\mathcal{D}$
for any ϵ $(where 0 \leq \epsilon < \frac{1}{2})$
for any δ $(where 0 \leq \delta < 1$
algorithm $\mathcal{L}$ on input ϵ and δ and a set of random examples picked from any probability distribution $\mathcal{D}$ outputs at least with a probability $1 - \delta$, concept h such that $error(h, f) \leq \epsilon$.

Theoretical Result: Learning Boolean formula even approximately is hard.
3-DNF is not learnable in Probably Approximately Correct framework unless RP = NP.

Two Key Ideas







$\phi_{explain}$:
Using explanation vocabulary
Ex: Obstacle presence

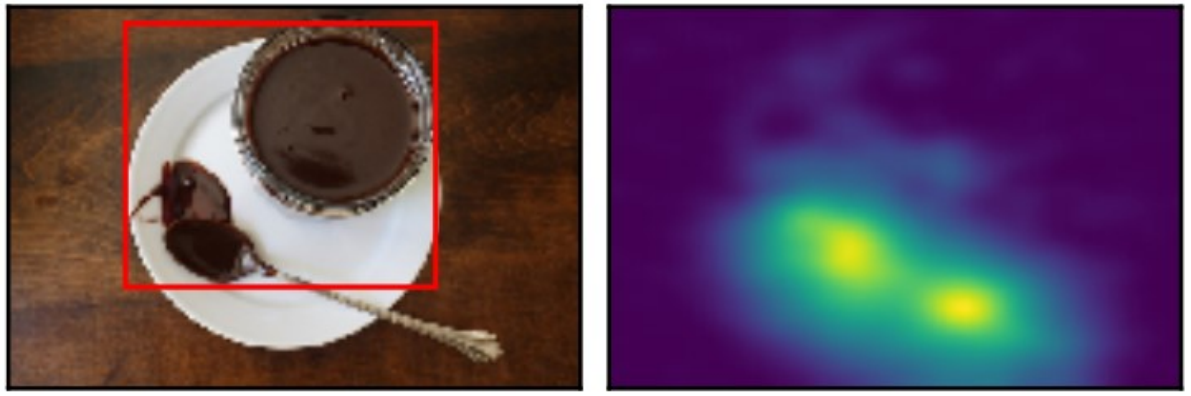
ϕ_{query} :
Some property of the output
Ex: Some cells not selected

1. Vocabulary is large.
   ıy samples (and what
   on) to consider for
   explanation ?
   Boolean formula with
   antees is hard.

Active learning Boolean formula $\phi_{explain}$ and not learning from fixed sample.

Explanations are often short and involve only few variables !

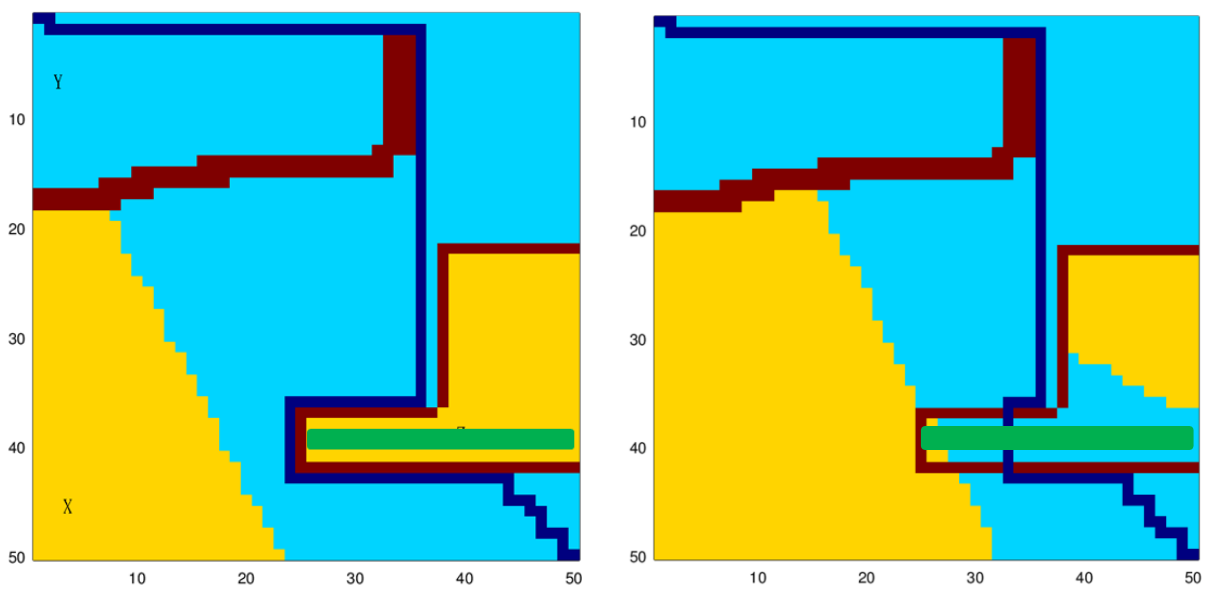
Two Key Ideas



Active learning Boolean formula $\phi_{explain}$ and not learning from fixed sample.

Explanations are often short and involve only few variables !

Two Key Ideas



Involves only two variables.
If we knew which two, we had only $2^{2^2} = 16$ possible explanations.

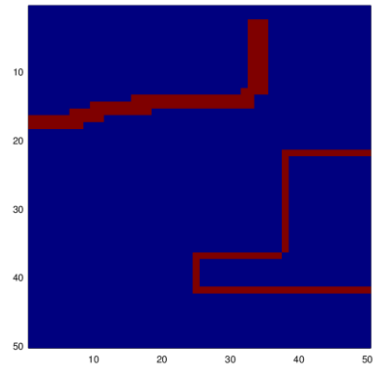
How do we find these relevant variables?

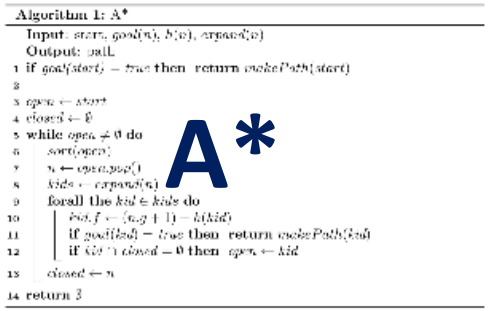
Active learning Boolean formula $\phi_{explain}$ and not learning from fixed sample.

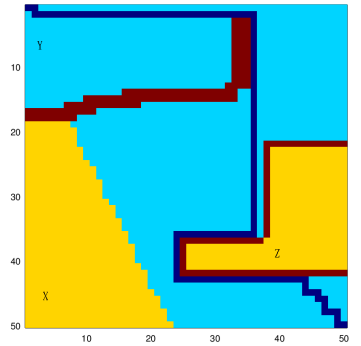
Explanations are often short and involve only few variables !

Actively Learning Boolean Formula

Oracle







Assignments to V
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

ϕ_{query} :
Some property of the output
Ex: Some cells not selected

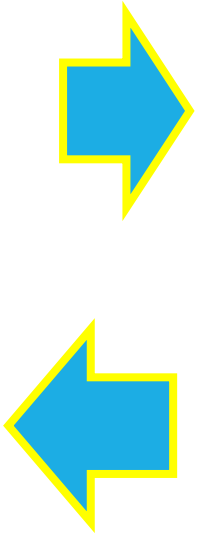
$\phi_{explain}$ (V) :
Using explanation vocabulary
Ex: Obstacle presence

ϕ
Evaluates assignments and returns T,F

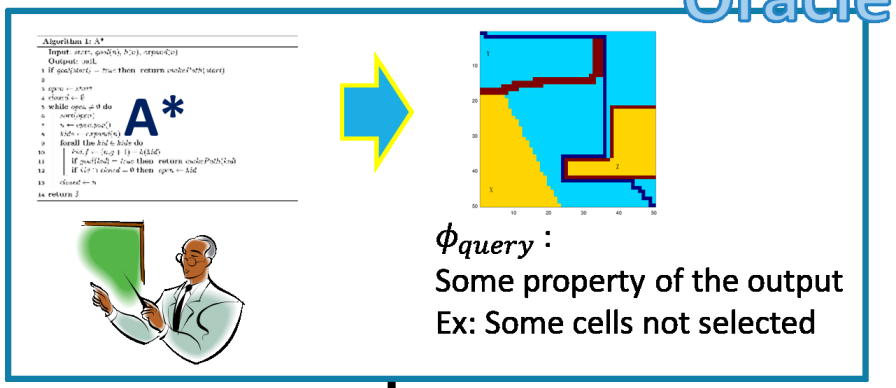
Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Assignments to V
m1 = (0,0,0,1,1,0,1)



Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1 : True

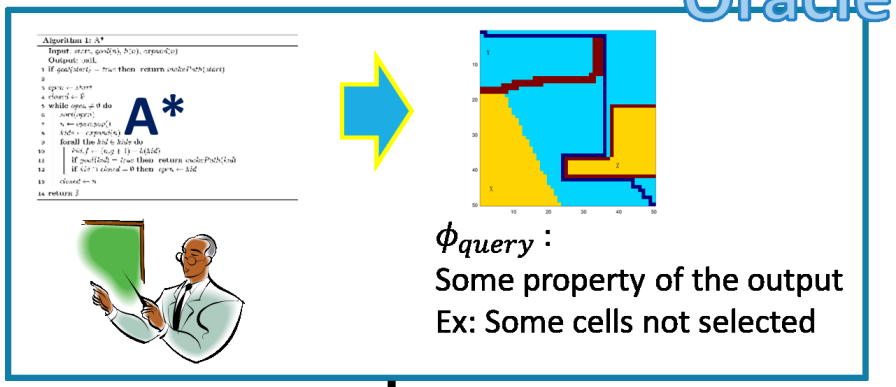
Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Assignments to V
m1 = (0,0,0,1,1,0,1)
m2 = (0,0,1,1,0,1,0)

Random Sample Till Oracle differs

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1: True, m2: False

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Assignments to V

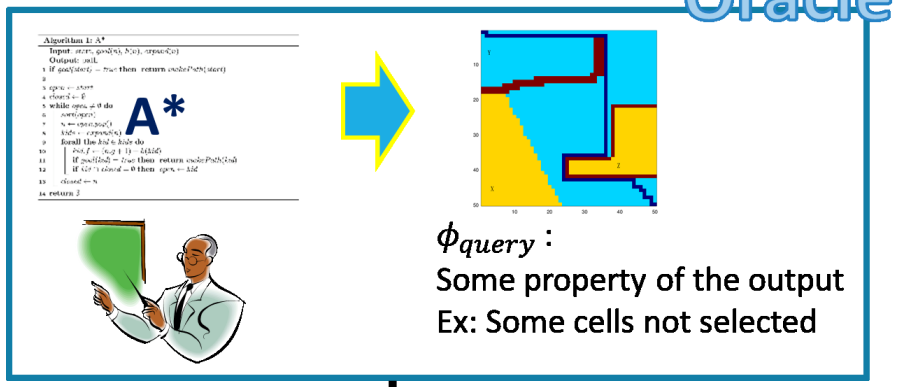
m1 = (0,0,0,1,1,0,1)

m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1: True, m2: False

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Assignments to V

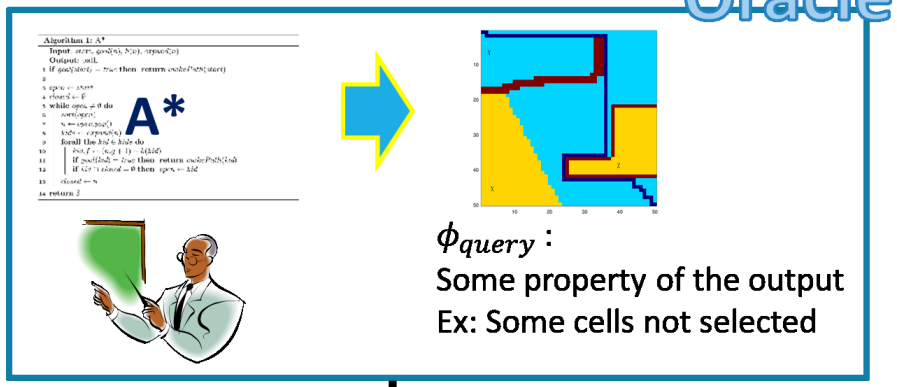
m1 = (0,0,0,1,1,0,1)

m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m1: True, m2: False

m3: True

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Hamming Distance = 4

Hamming Distance = 2

Assignments to V

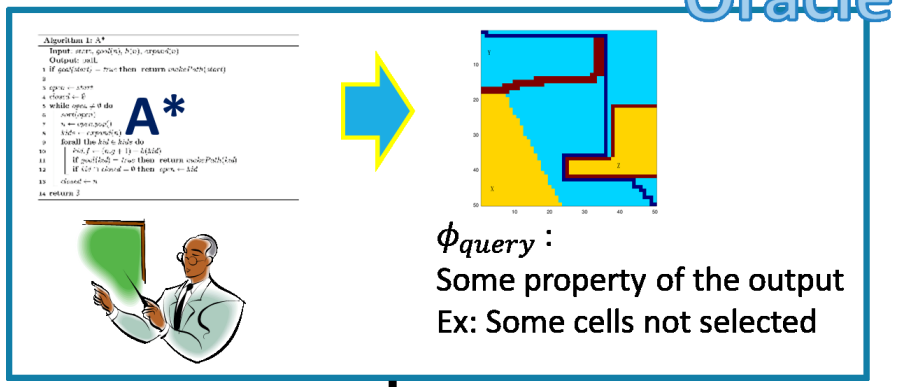
~~m1 = (0,0,0,1,1,0,1)~~

m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

Binary Search Over Hamming Distance

Oracle



~~m1: True,~~ m2: False

m3: True

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Hamming Distance = 2

Hamming Distance = 1

Assignments to V

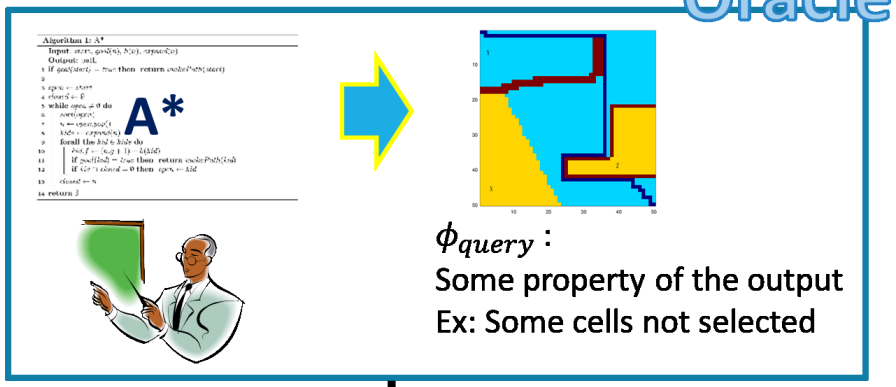
m2 = (0,0,1,1,0,1,0)

m3 = (0,0,0,1,1,1,0)

m4 = (0,0,1,1,1,1,0)

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m2: False, m3: True

m4: True

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Hamming Distance = 2

Hamming Distance = 1

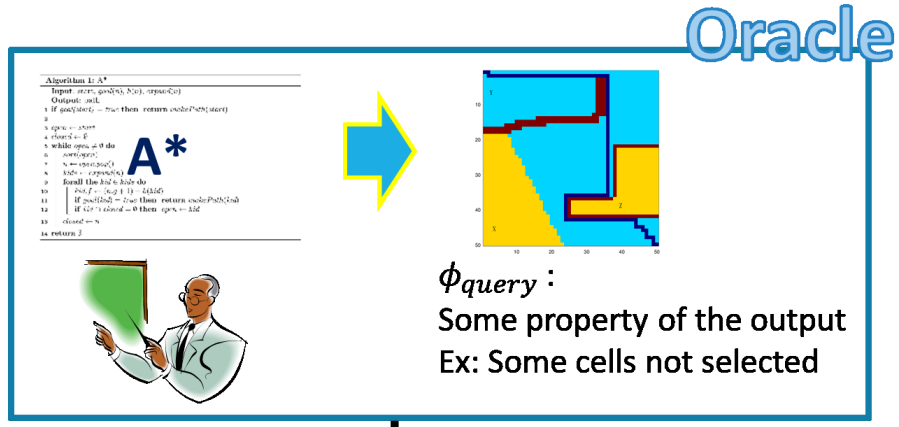
Assignments to V

m2 = (0,0,1,1,0,1,0)

~~m3 = (0,0,0,1,1,1,0)~~

m4 = (0,0,1,1,1,1,0)

Binary Search Over Hamming Distance



m2: False, ~~m3: True~~

m4: True

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Hamming Distance = 1

Assignments to V

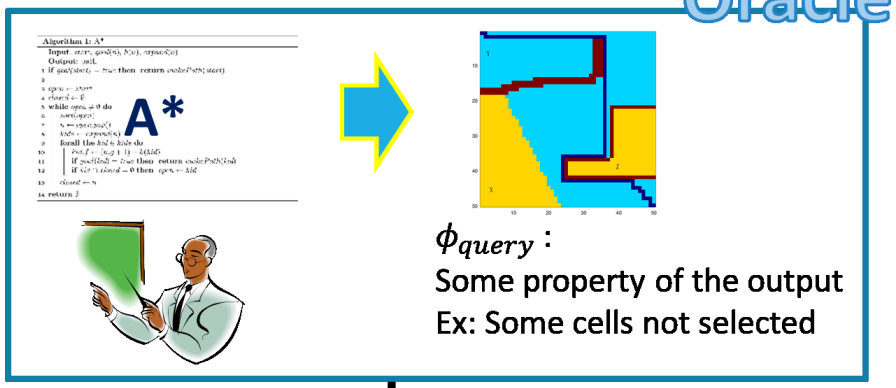
m2 = (0,0,1,1,0,1,0)

m4 = (0,0,1,1,1,1,0)

Fifth variable v_5 is relevant !!

Binary Search Over Hamming Distance

Oracle



ϕ_{query} :
Some property of the output
Ex: Some cells not selected

m2: False, m4: True

Actively Learning Relevant Variables

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

For each assignment to relevant variables

$$2^{|U|}$$

Random Sample Till Oracle differs

$$ln(1/(1-\kappa))$$

Binary Search Over Hamming Distance

$$ln(|V|)$$

Relevant variables of $\phi_{explain}$ found with confidence κ in

$$\mathbf{2^{|U|} ln(\frac{|V|}{1-\kappa})}$$

Actively Learning Boolean Formula

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

Used distinguishing input based approach from ICSE'10

Susmit Jha, Sumit Gulwani, Sanjit A Seshia, and Ashish Tiwari. Oracle-guided component-based program synthesis. In *2010 ACM/IEEE 32nd International Conference on Software Engineering*, volume 1, pages 215–224. IEEE, 2010.

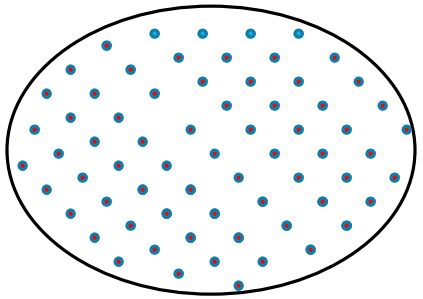
Build Truth Table for the relevant variables U

$Worst\ Case\colon\ 2^{|U|}$

$\phi_{explain}$ found with confidence κ in

$$\mathbf{2^{|U|}\left(1 + ln\left(\frac{|V|}{1-\kappa}\right)\right)}$$

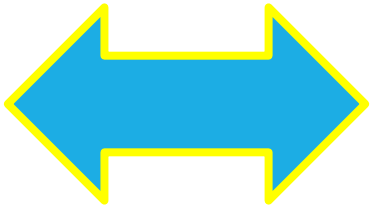
Distinguishing Example



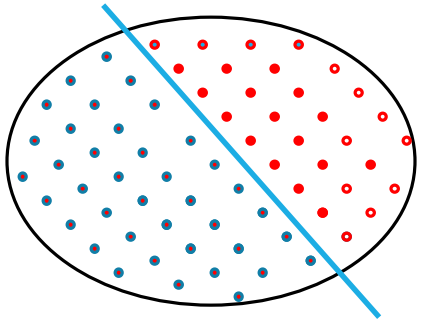
Space of all possible $2^{2^{|U|}}$ formulae.
Each dot represents semantically unique Boolean formula

Distinguishing Example

Oracle O: Label e_1 as true iff it satisfies target ϕ



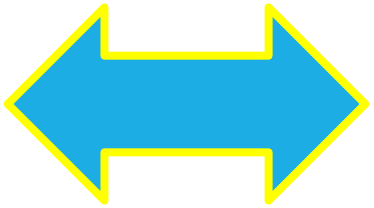
Randomly select - assignment to d Boolean variables - e_1



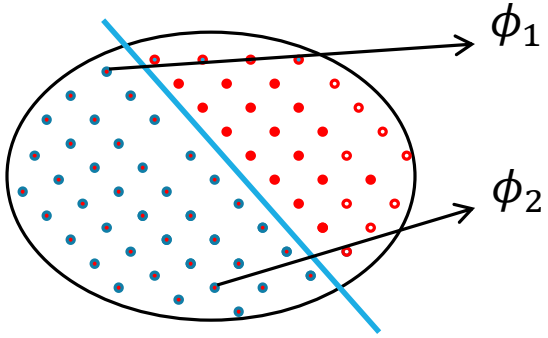
some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1\}$

Distinguishing Example

Oracle O: Label e_1 as true iff it satisfies target ϕ



Randomly select - assignment to d Boolean variables - e_1

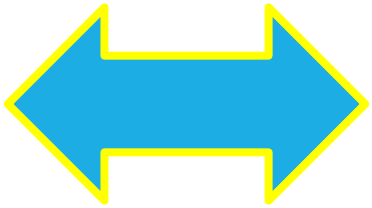


2 Candidate Formulae Consistent with Example Set

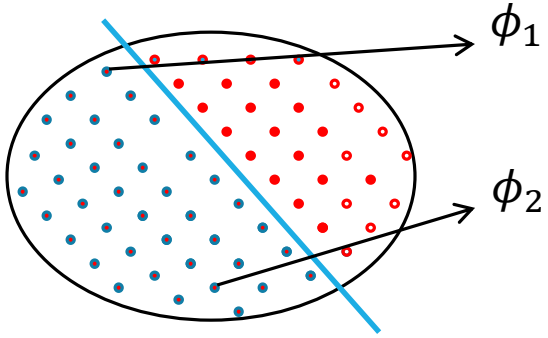
some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1\}$

Distinguishing Example

Oracle O: Label e_2 as true iff it satisfies target ϕ



Randomly select - assignment to d Boolean variables - e_1



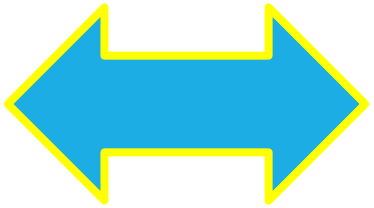
Find distinguishing example

$e_2 \vDash \phi_1 \oplus \phi_2$

some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1\}$

Distinguishing Example

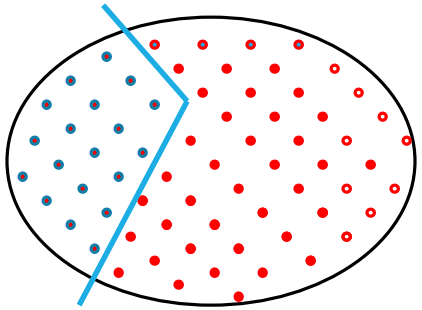
Oracle O: Label e_2 as true iff it satisfies target ϕ



Randomly select - assignment to d Boolean variables - e_1

Find distinguishing example

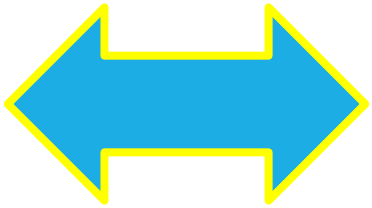
$e_2 \vDash \phi_1 \oplus \phi_2$



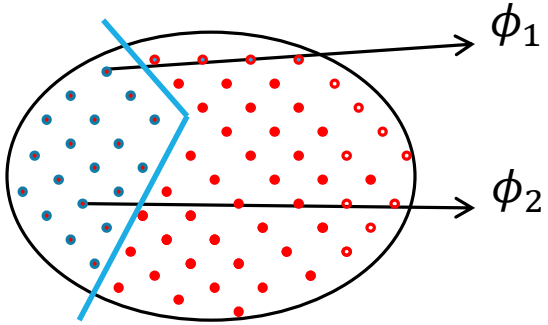
some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1, e_2\}$

Distinguishing Example

Oracle O: Label e_2 as true iff it satisfies target ϕ



Randomly select - assignment to d Boolean variables - e_1



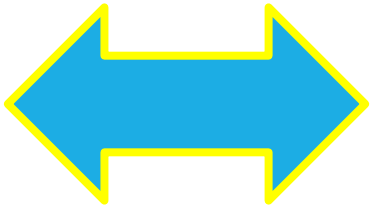
Find distinguishing example

$e_3 \vDash \phi_1 \oplus \phi_2$

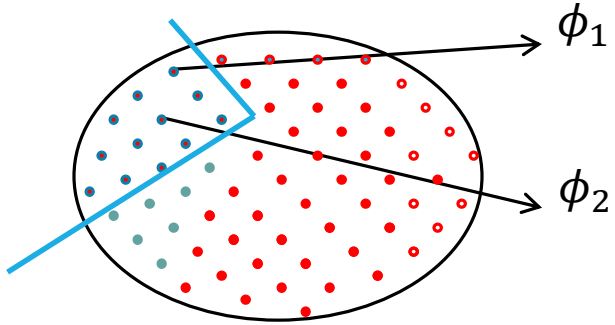
some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1, e_2\}$

Distinguishing Example

Oracle O: Label e_2 as true iff it satisfies target ϕ



Randomly select - assignment to d Boolean variables - e_1



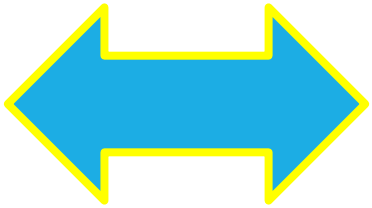
Find distinguishing example

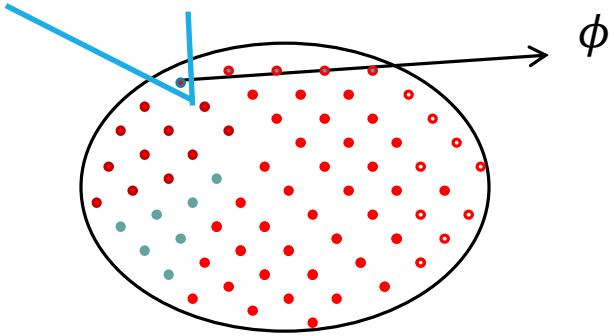
$e_3 \vDash \phi_1 \oplus \phi_2$

some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1, e_2, e_3\}$

Distinguishing Example

Oracle O: Label e_2 as true iff it satisfies target ϕ





some of the $2^{2^{|U|}}$ formula are not consistent with example set $\{e_1, e_2, e_3, \ldots, e_k\}$

Randomly select - assignment to d Boolean variables - e_1

Find distinguishing example $e_k \vDash \phi_1 \oplus \phi_2$

Terminate when we are left with a single consistent ϕ w.r.t all the examples.

Actively Learning Boolean Formula

$$Find\ U\ such\ that\ \phi_{explain}(\mathrm{V}) \equiv \phi_{explain}(U)\ where\ |U| \ll |V|$$

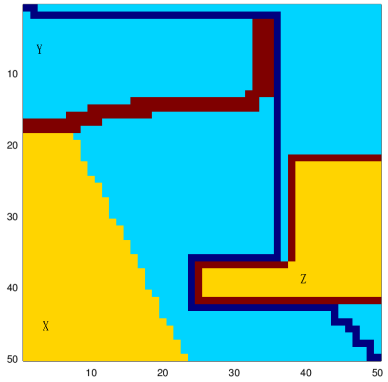
Correctness Guarantee: If there exists an explanation using variables in V, then this approach is guaranteed to find it (learn correct Boolean formula).

If there does not exist an explanation using V, the approach will either detect it or find a wrong explanation (learn incorrect Boolean formula).

In practice, since we depend only logarithmically on $|\mathrm{V}|$, we use large vocabulary.

$$\mathbf{2^{|U|}\left(1 + ln\left(\frac{|V|}{1-\kappa}\right)\right)}$$

Experiments



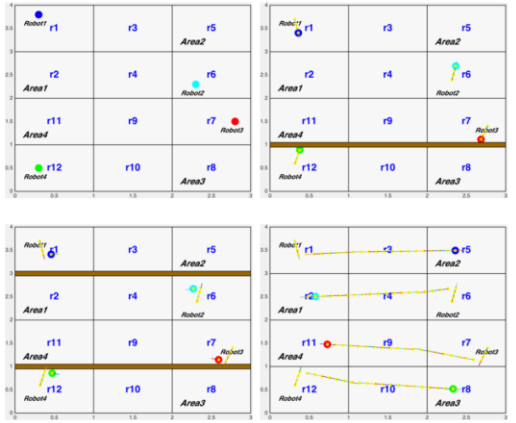
Explaining A* Planning

|V| = 2500

|U| <= 4

Runtime < 3 minutes

10^153



Reactive Exploration Strategy

|V| = 96

|U| <= 2

Runtime < 5 seconds

10^28

Image Classification

|V| = 784

|U| <= 27

Runtime < 5 minutes

Challenges

Improving selection of inputs to run to reduce number of introspections needed.

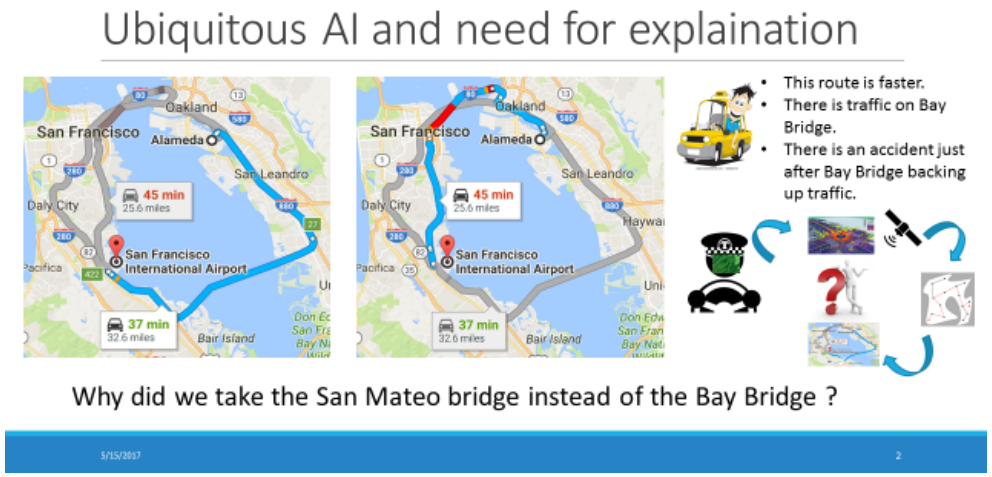
- Exploit structural assumption on nature of explanation
- Exploit probability distribution over inputs

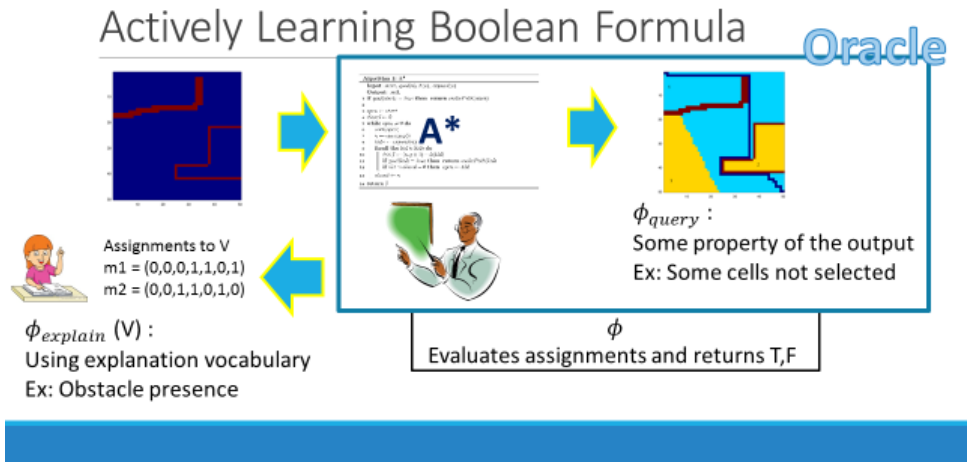
Exploitation of partial knowledge about AI algorithms to enable efficient sampling.

Extension to randomized decision making algorithms which may make different decisions for the same input.

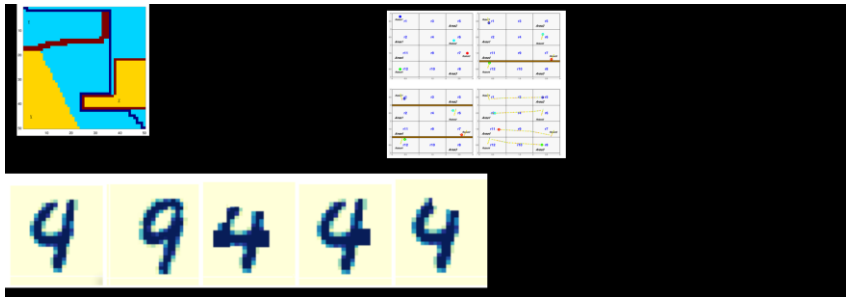
Automated/Iterative Selection of Vocabulary

Thanks!!





$\phi_{explain}$ found with confidence κ in $O(2^{|U|} ln(|V|/(1-\kappa)))$



Thanks!

It is really confusing!!!