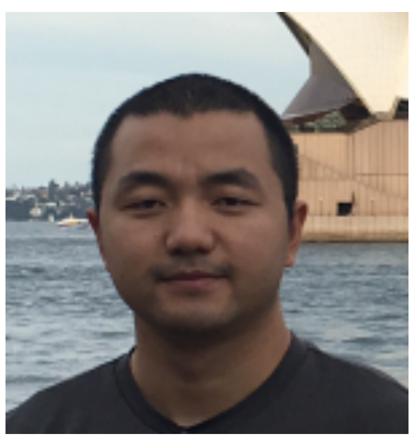
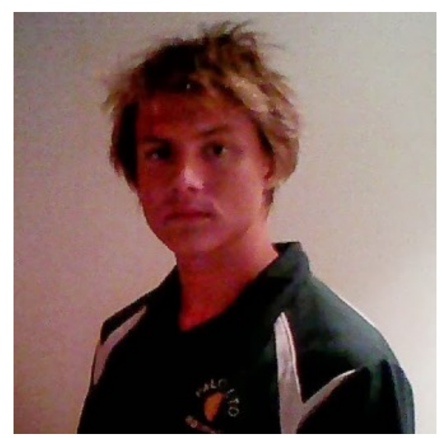
Efficient Nonmyopic Batch Active Search



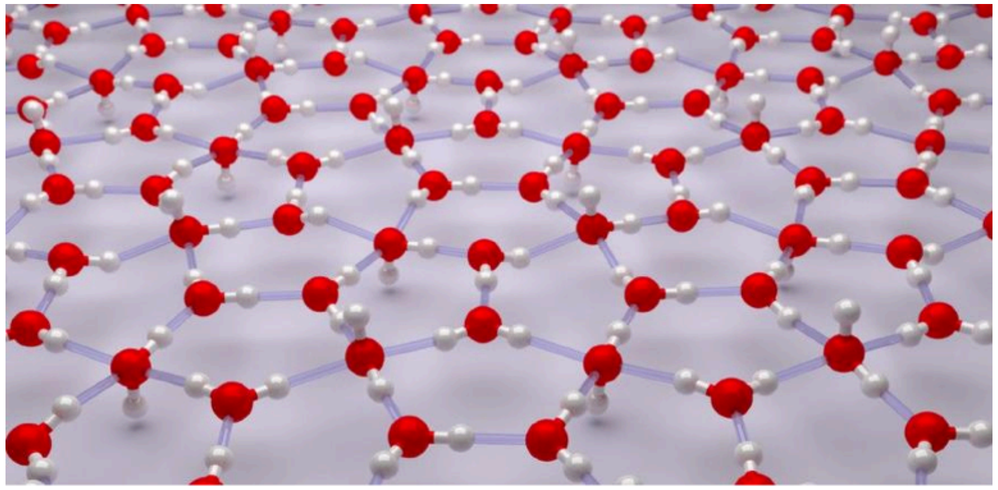


Shali Jiang Gustavo Malkomes Matthew Abbott Benjamin Moseley Roman Garnett

NeurIPS 2018

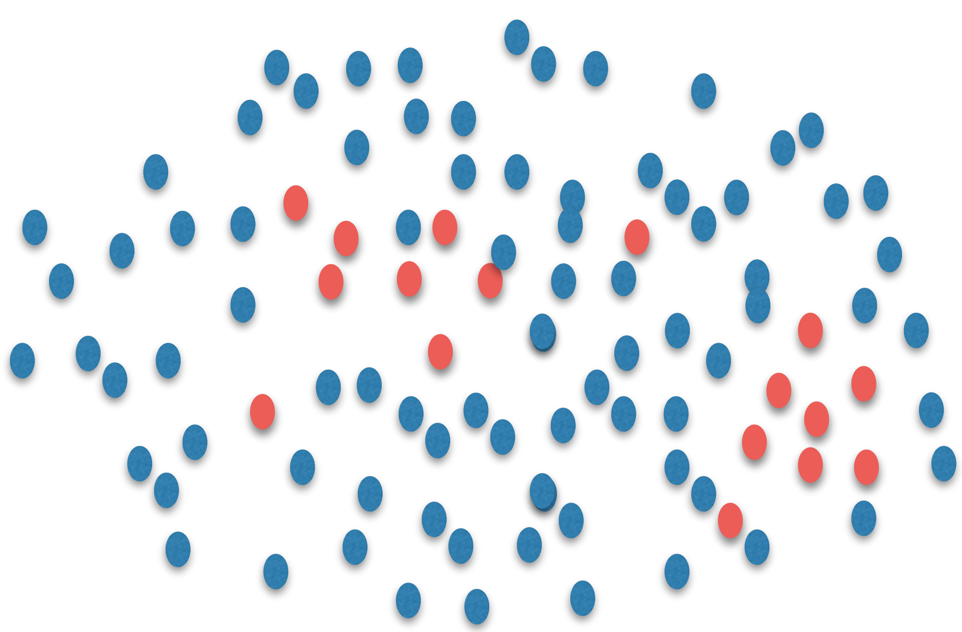
Many real problems involve searching for valuable items from a large pool of candidates in an iterative fashion

Drug discovery

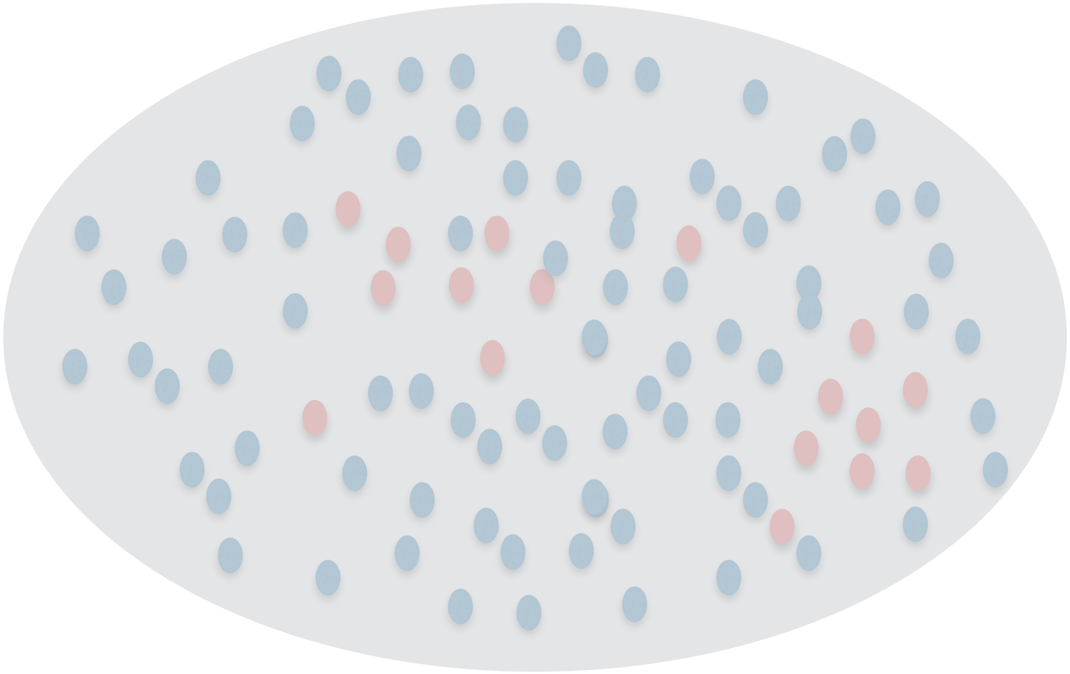


Materials discovery

What is active search

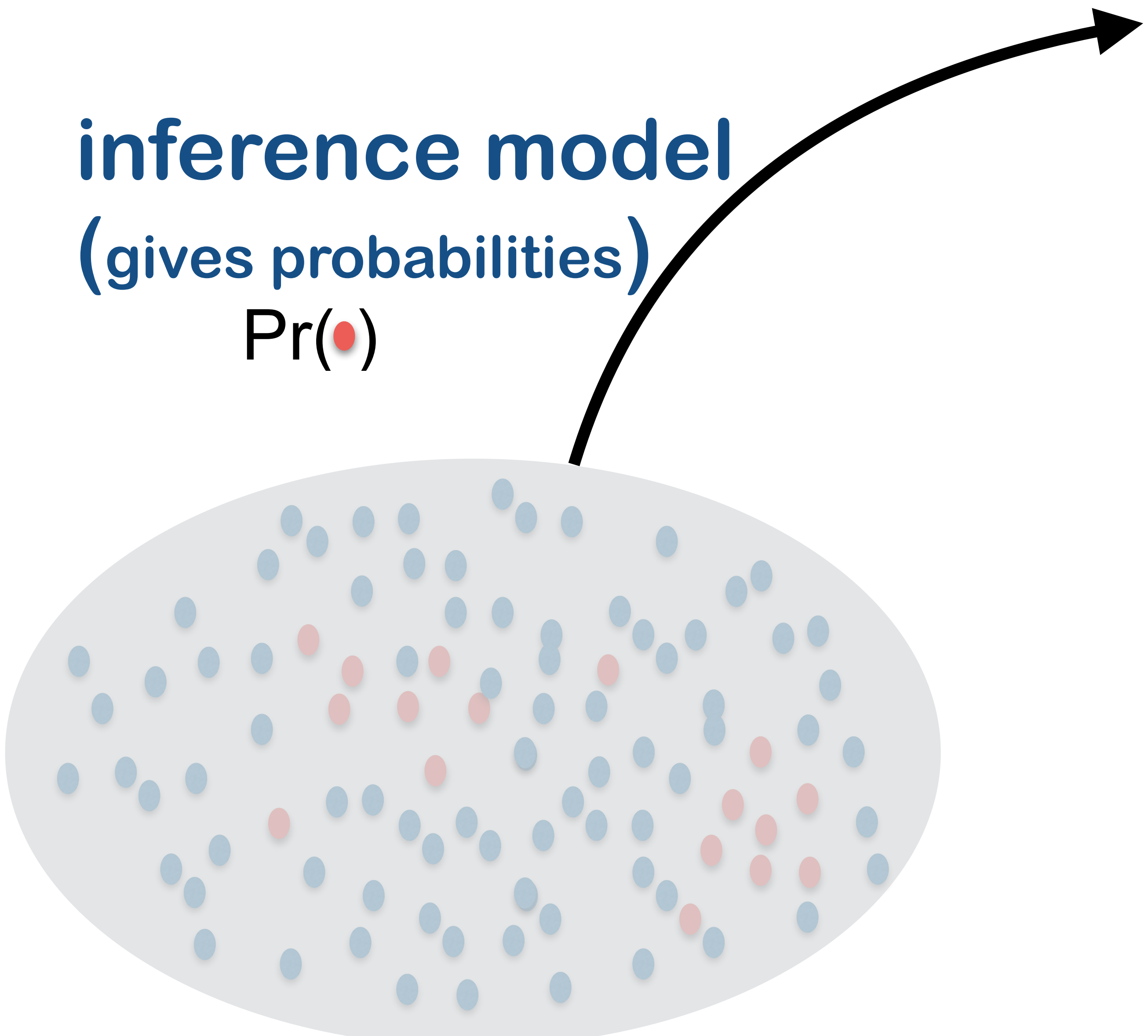


What is active search

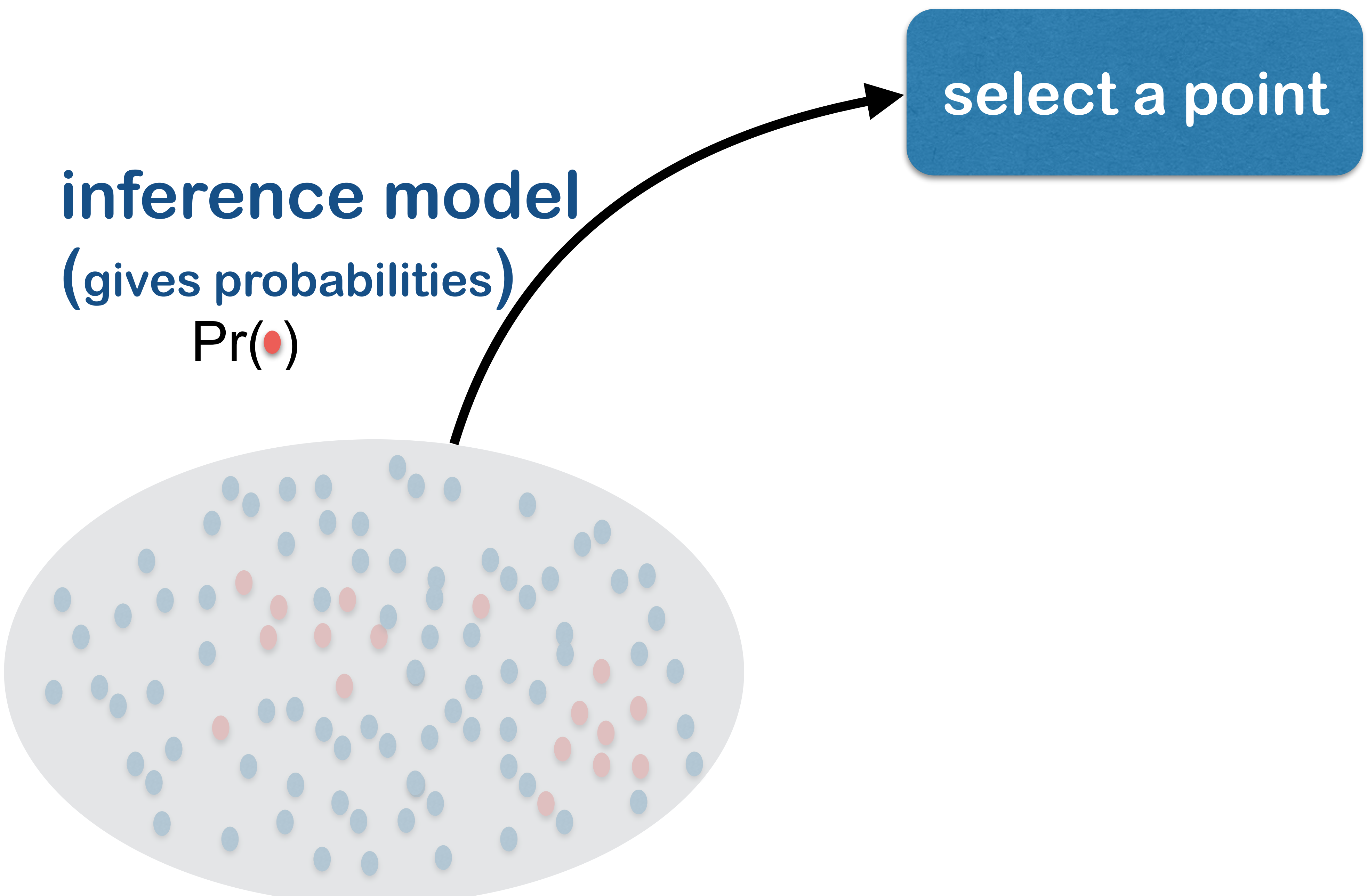


What is active search

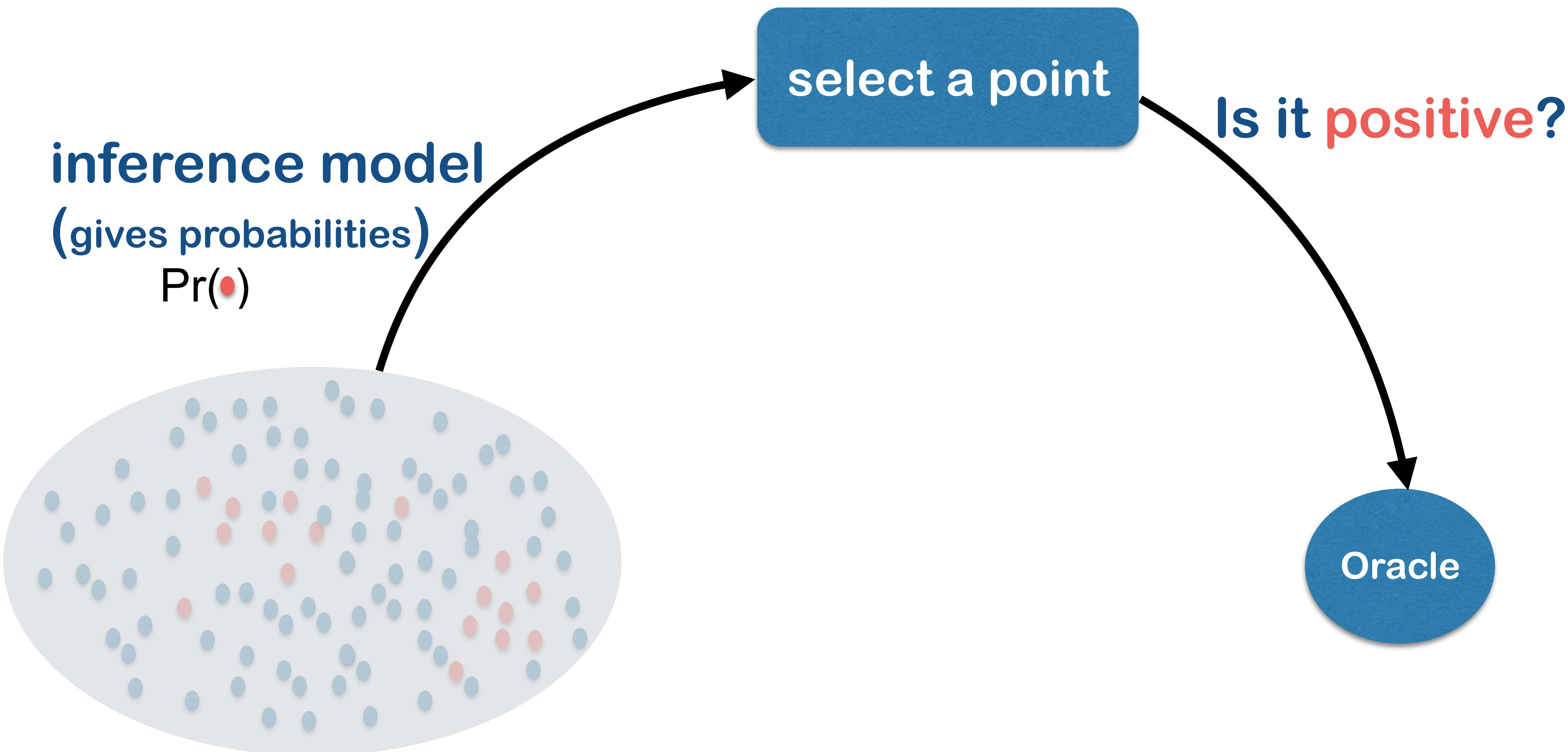
inference model
(gives probabilities)



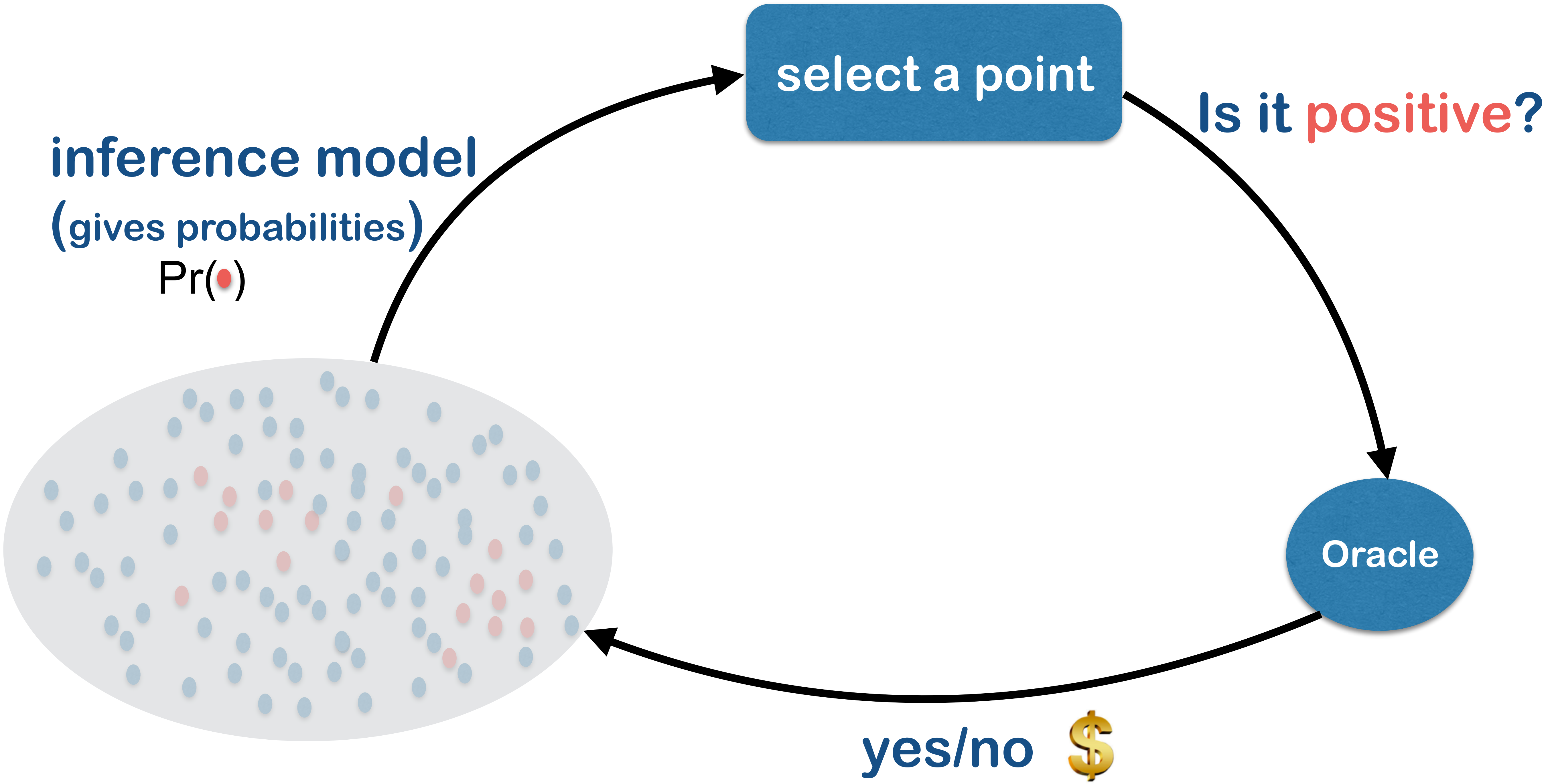
What is active search



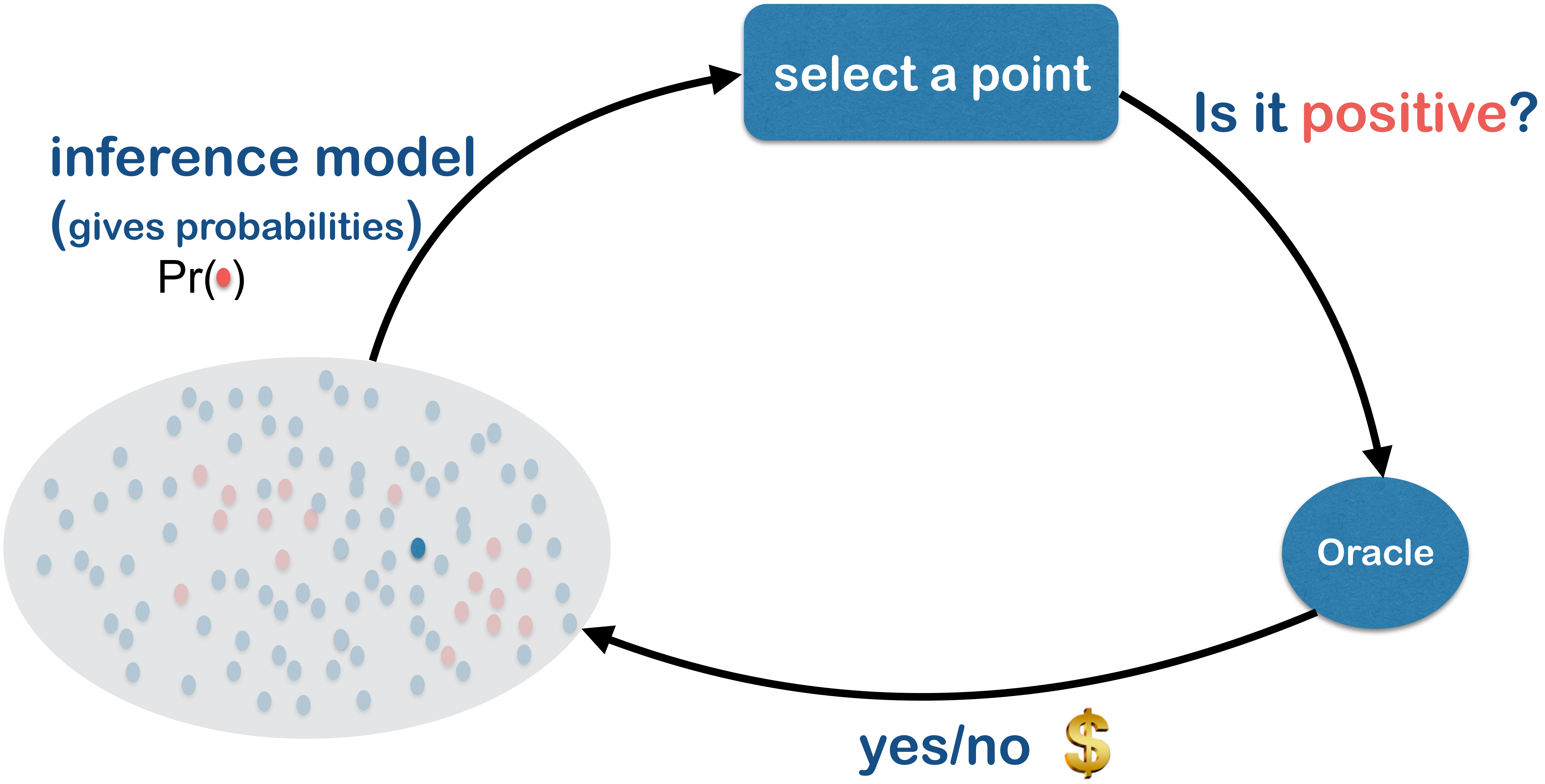
What is active search



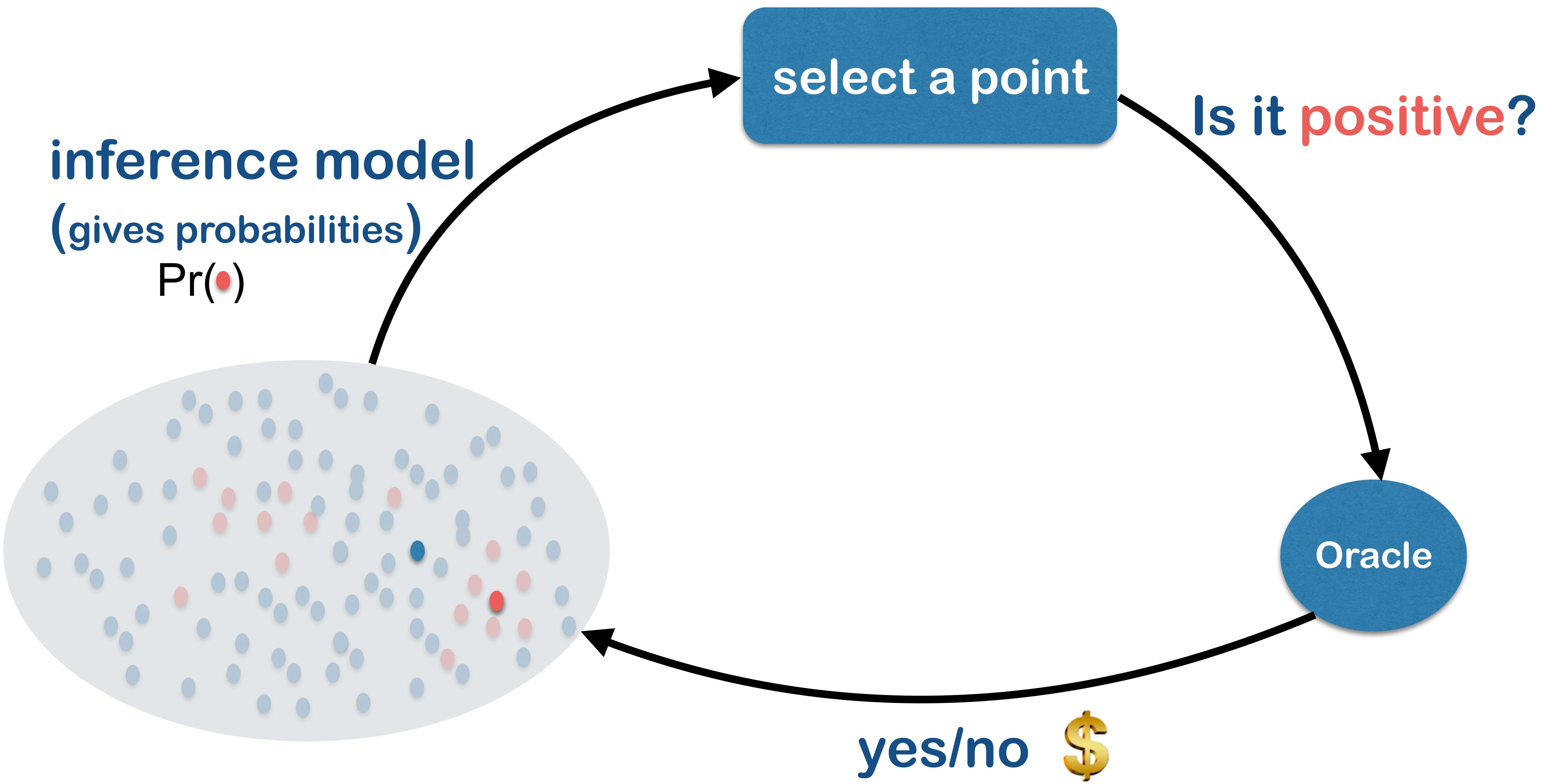
What is active search



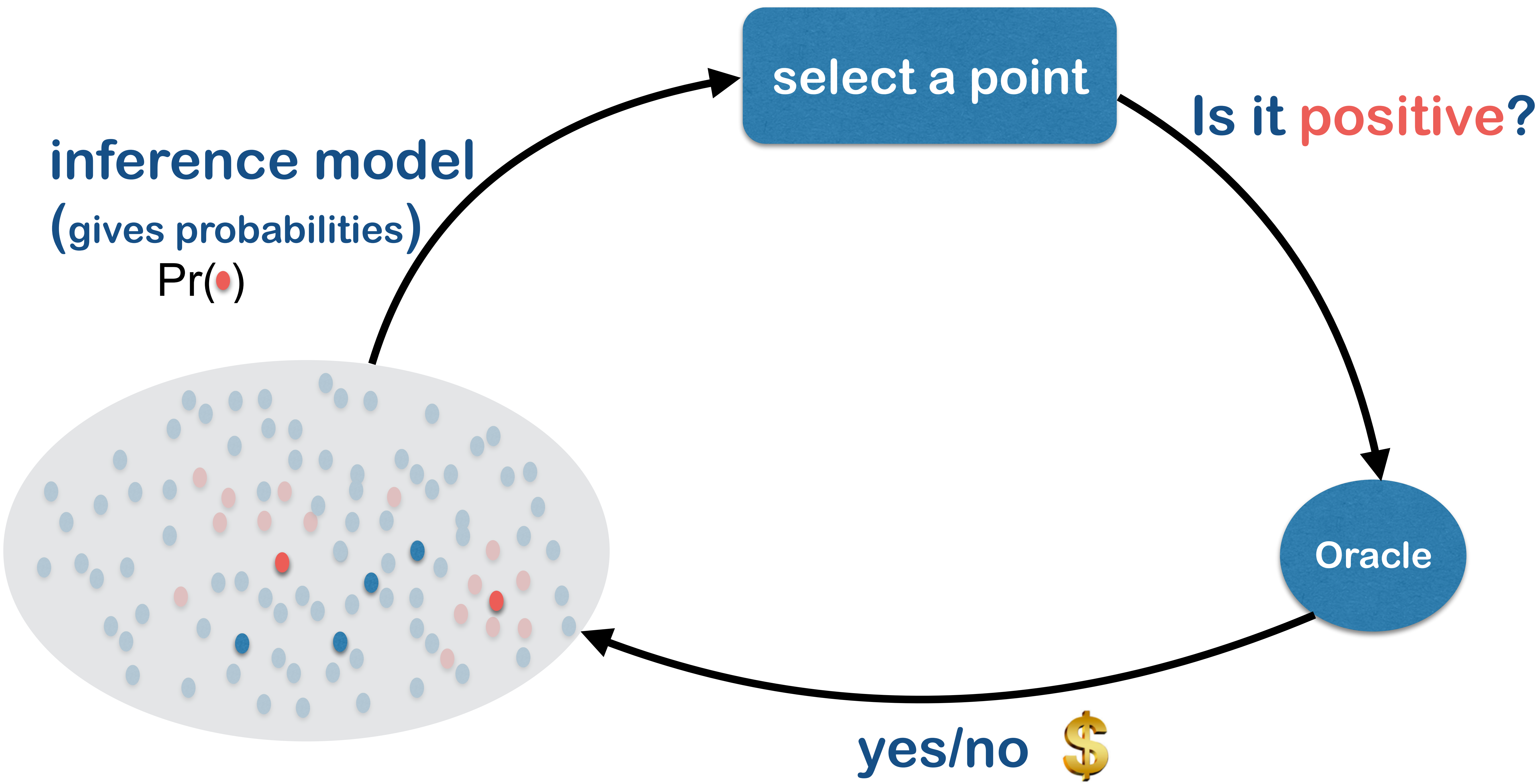
What is active search



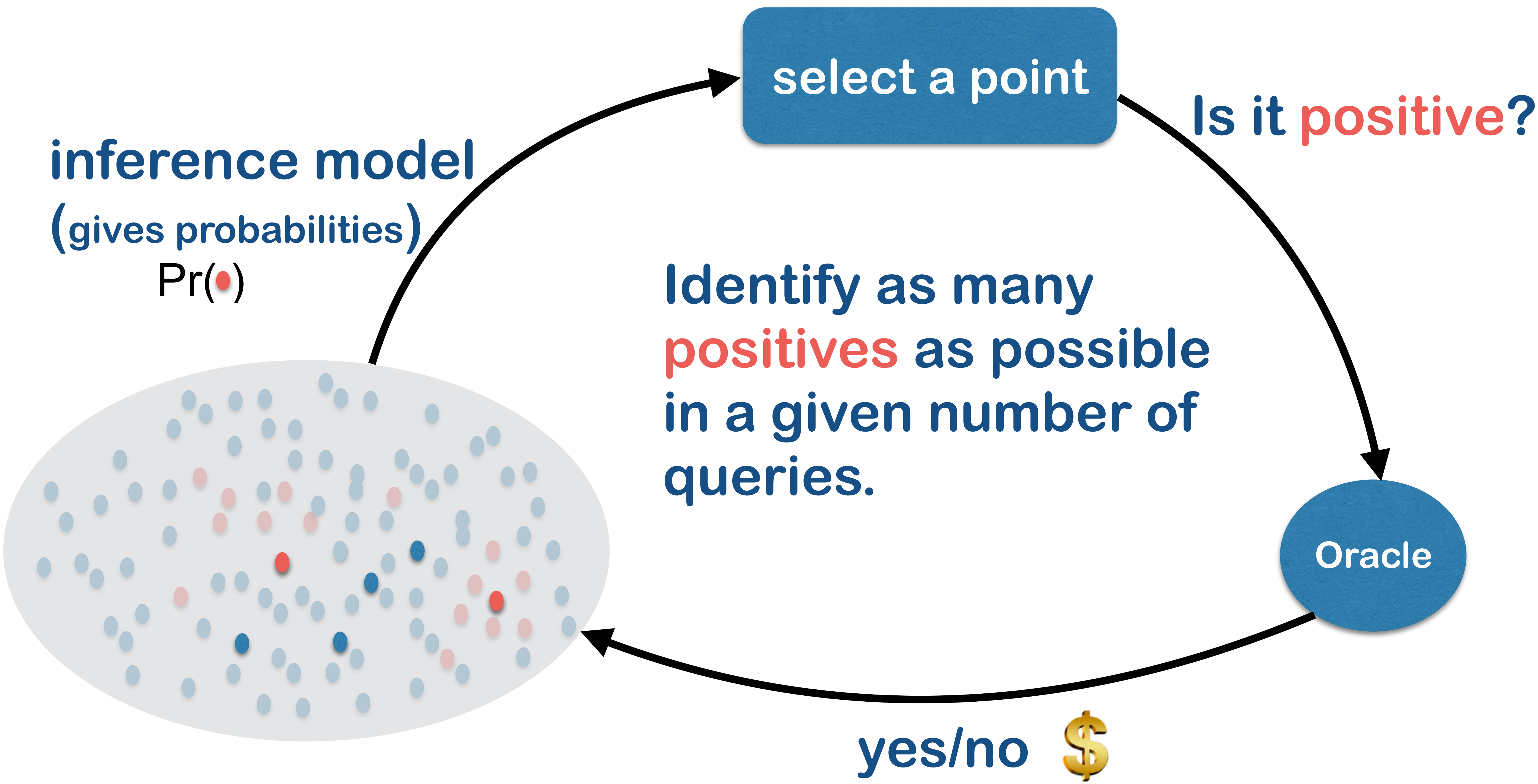
What is active search



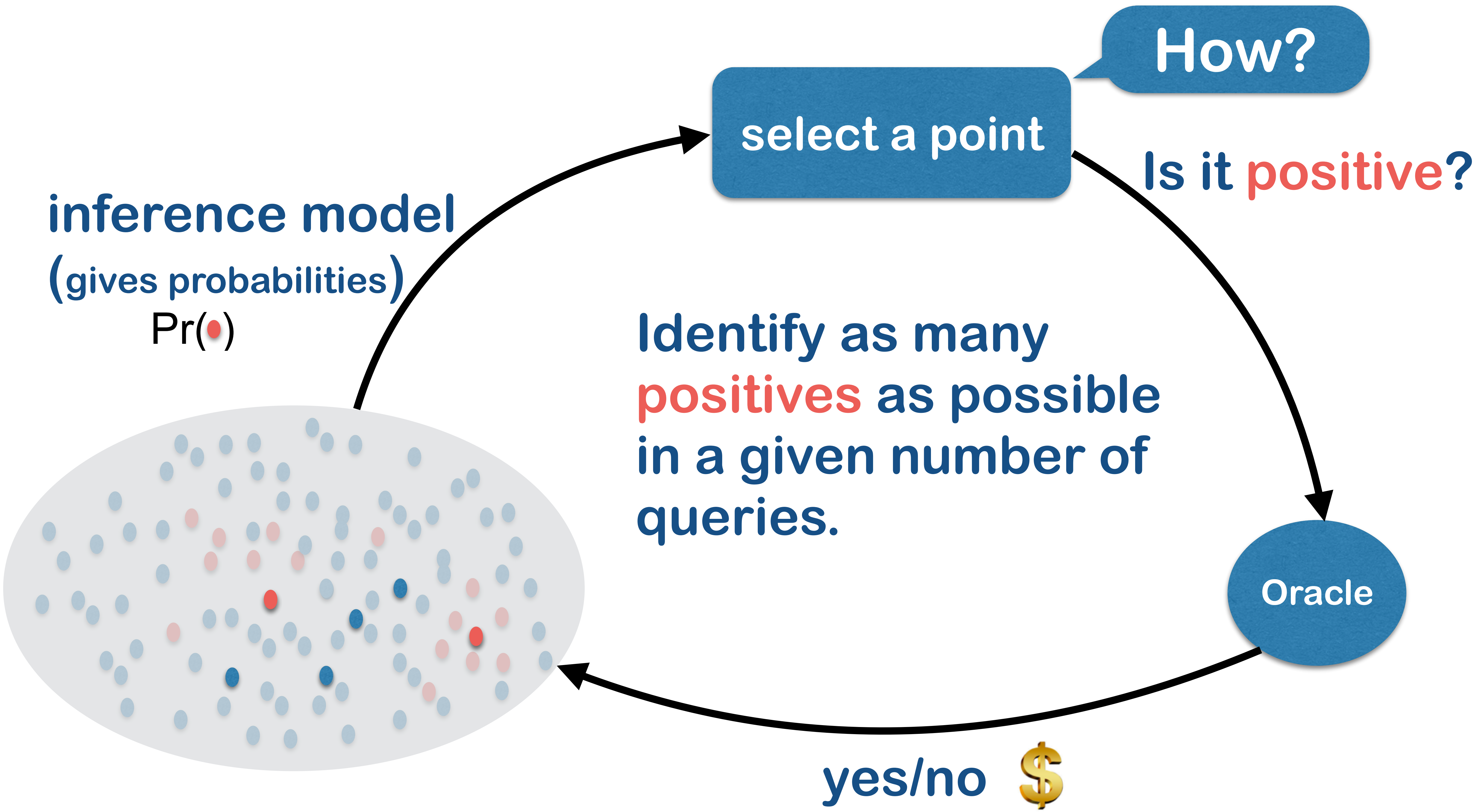
What is active search



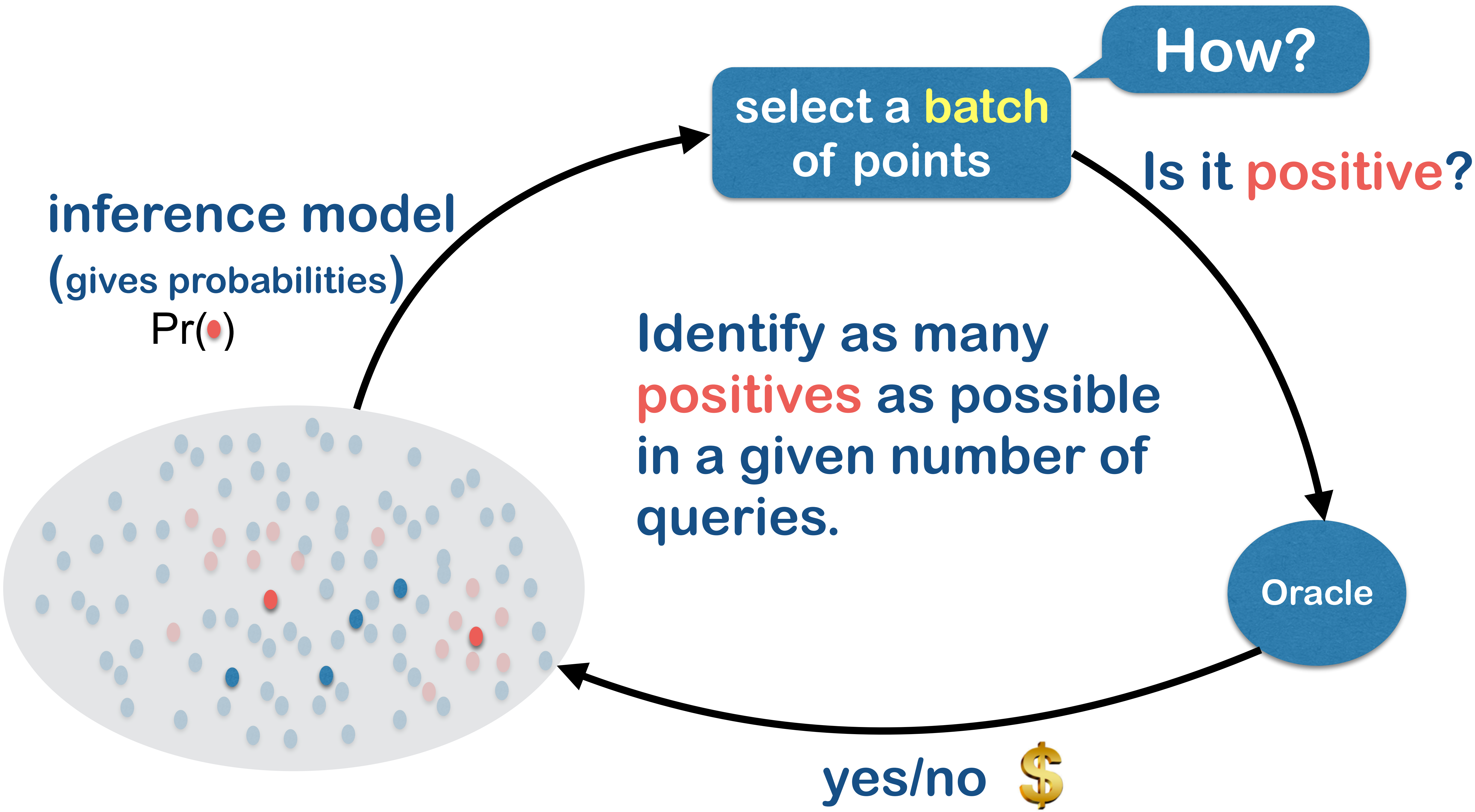
What is active search



What is active search



What is active search



The most straightforward policy one might think of is **greedy**:

—always choose the points with highest probabilities

The most straightforward policy one might think of is **greedy**:

—always choose the points with highest probabilities

$$X^* = \arg\max_X \text{“expected \#positives in } X\text{”}$$

The most straightforward policy one might think of is **greedy**:

—always choose the points with highest probabilities

$$X^* = \arg\max_X \text{“expected \#positives in } X\text{”}$$

It's efficient but myopic, ignoring what could happen in future

The most straightforward policy one might think of is greedy:

—always choose the points with highest probabilities

$$X^* = \arg\max_X \text{"expected \#positives in } X\text{"}$$

It's efficient but myopic, ignoring what could happen in future

How can we do better?

Nonmyopic: consider not only this batch, but also what could happen afterwards!

$$X^* = \arg\max_X \text{``[expected \#positives in } X\text{]+}$$

$$\boxed{\text{[expected \#positives in future conditioned on } X\text{]''.}}$$

Nonmyopic: consider not only this batch, but also what could happen afterwards!

$$X^* = \arg\max_X \text{“[expected \#positives in } X\text{]+}$$

$$\boxed{\text{[expected \#positives in future conditioned on } X\text{]”}.}$$

Assume conditional independence after X

Nonmyopic: consider not only this batch, but also what could happen afterwards!

$$X^* = \arg\max_X \text{``[expected \#positives in } X] +$$

$$\boxed{\text{[expected \#positives in future conditioned on } X]\text{''}.}$$

Assume conditional independence after X

Efficient for sequential setting (batch size 1) (Jiang et al. (ICML 2017)).

Nonmyopic: consider not only this batch, but also what could happen afterwards!

$$X^* = \arg\max_X \text{“[expected \#positives in } X\text{]+}$$

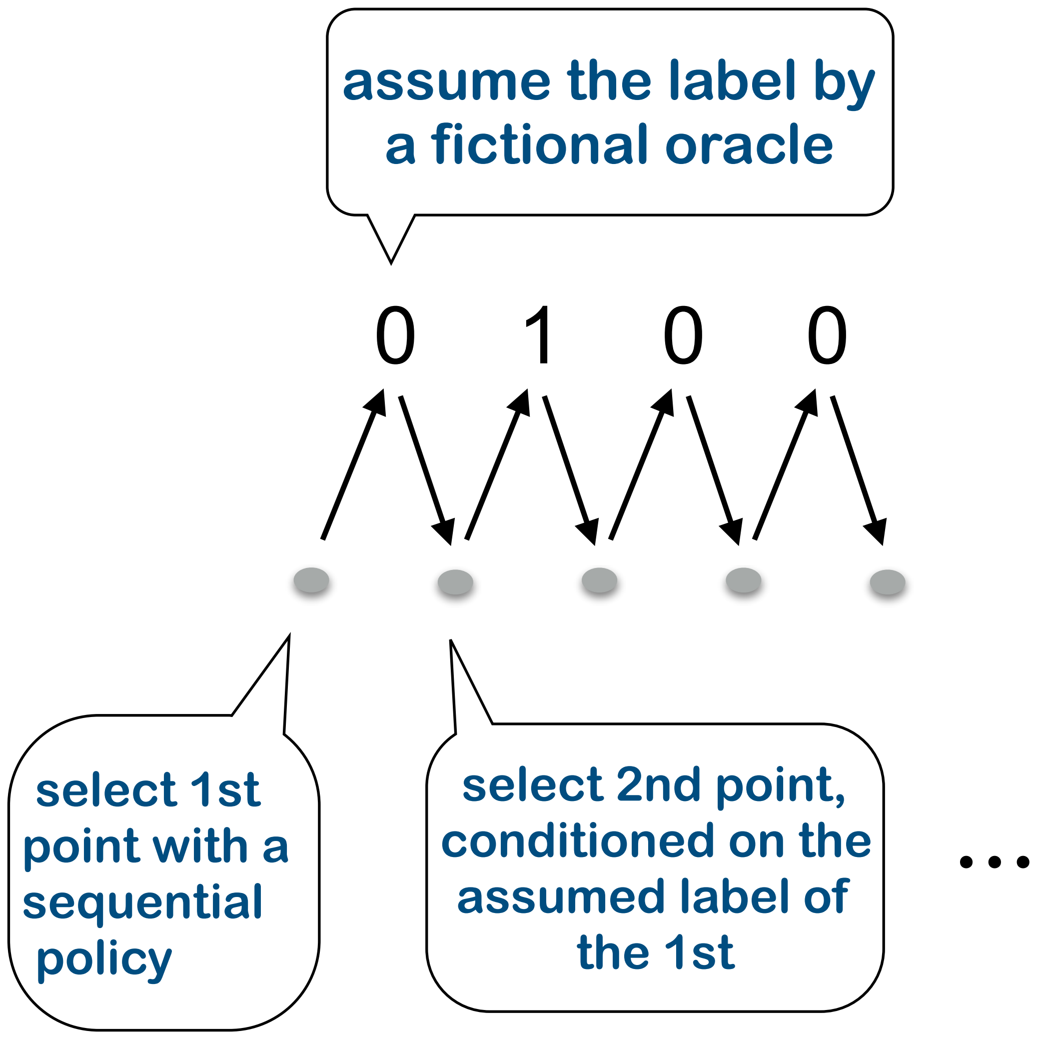
$$\boxed{\text{[expected \#positives in future conditioned on } X\text{]”.}}$$

Assume conditional independence after X

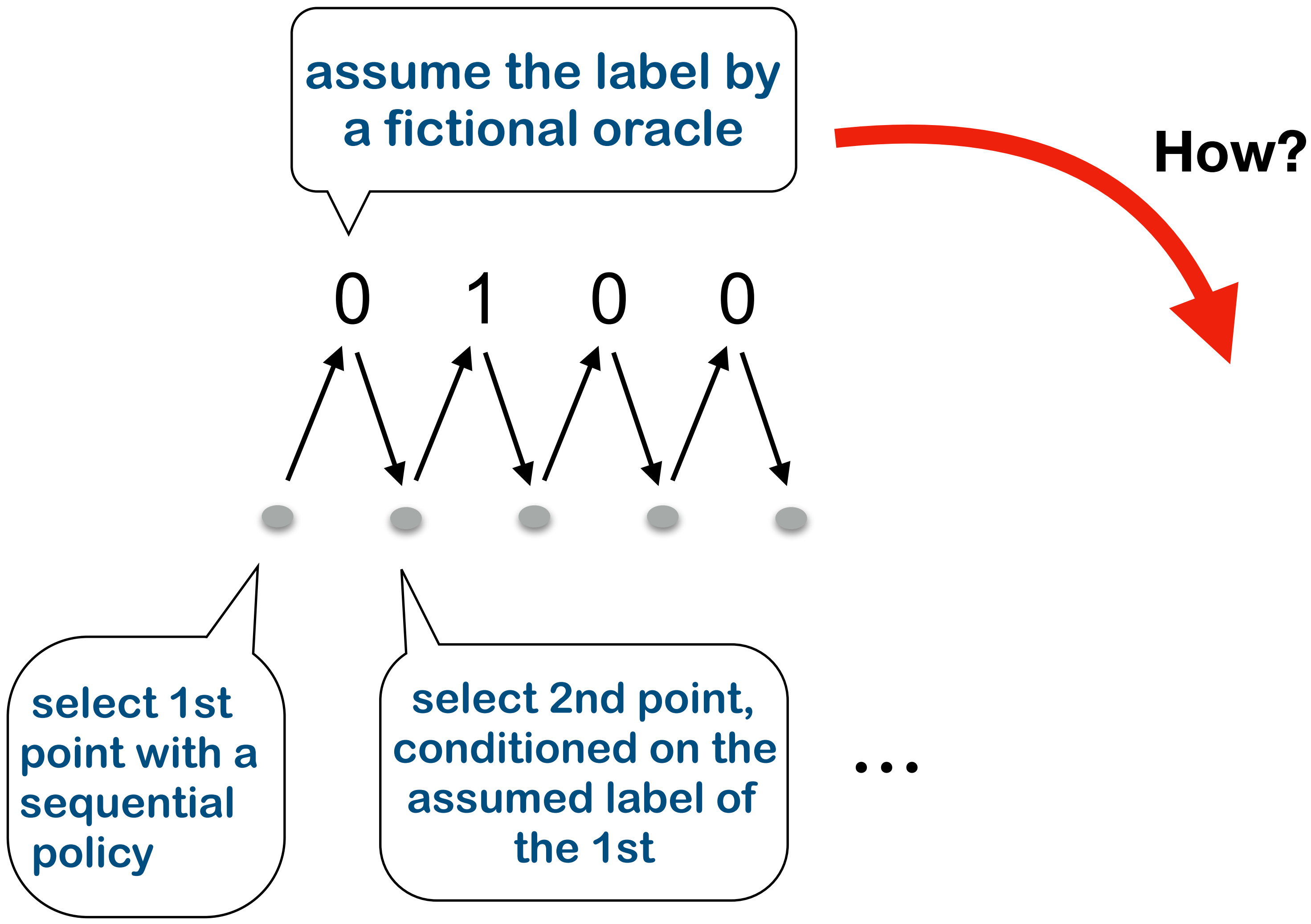
Efficient for sequential setting (batch size 1) (Jiang et al. (ICML 2017)).

Combinatorial search in batch setting → two approaches: greedy maximization and sequential simulation

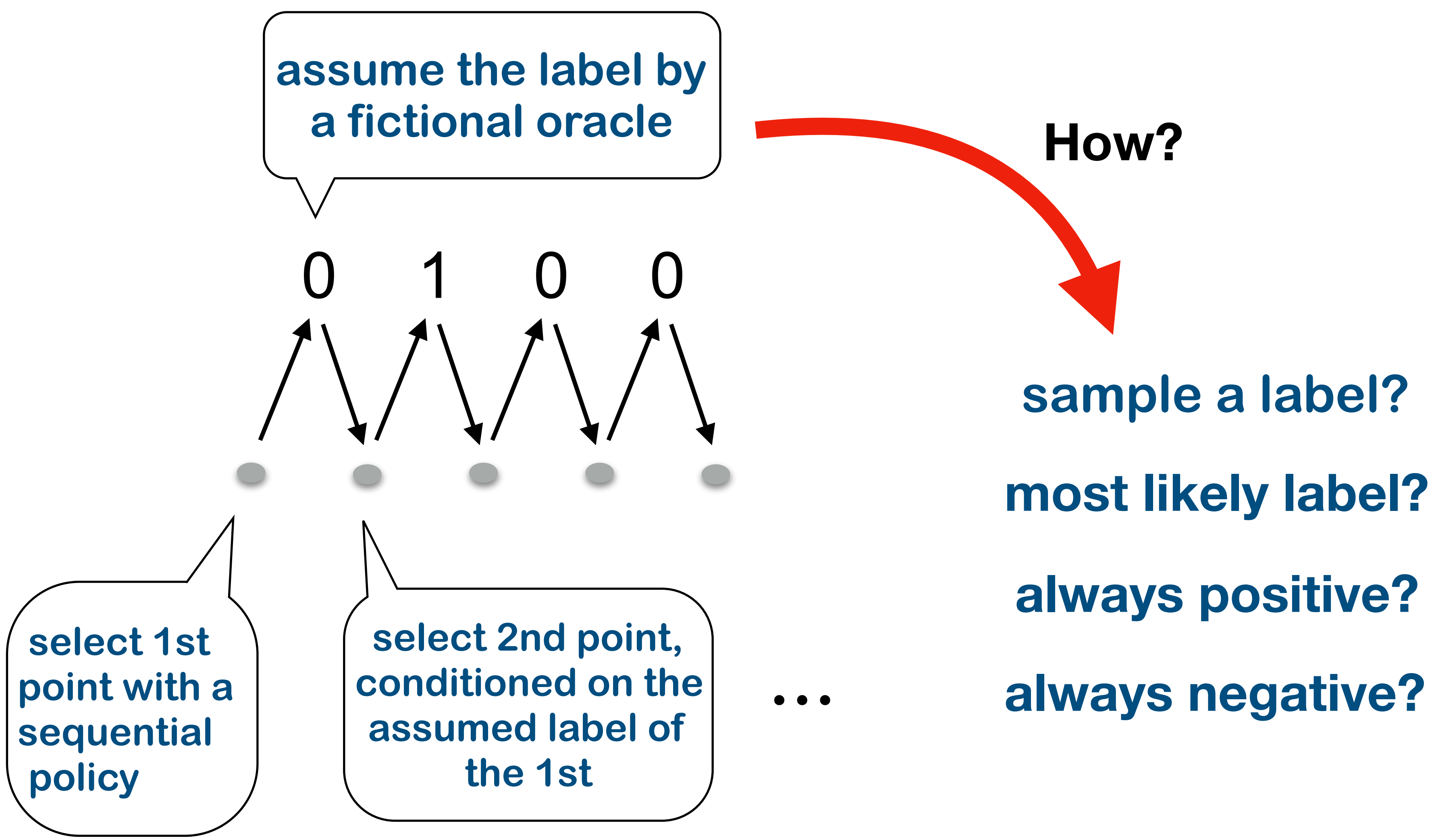
Sequential simulation



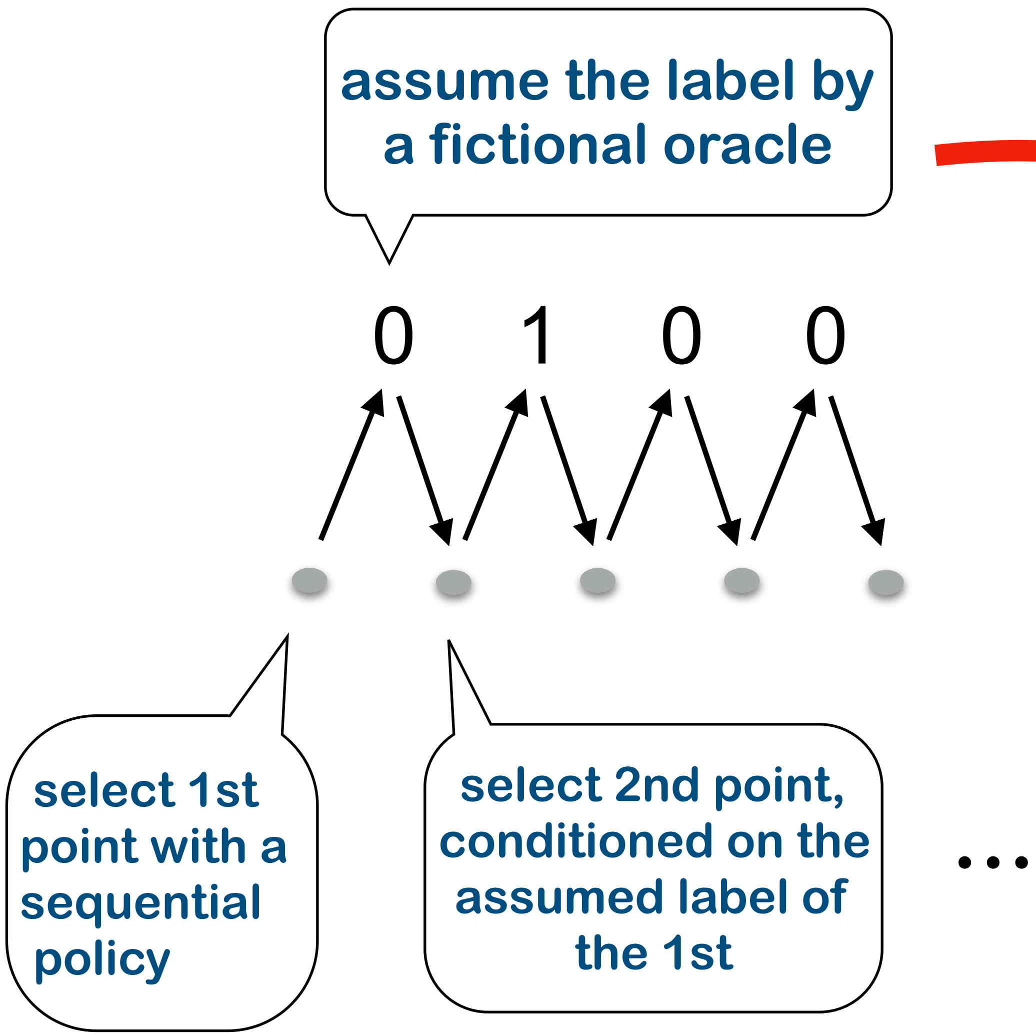
Sequential simulation



Sequential simulation



Sequential simulation



How?

sample a label?

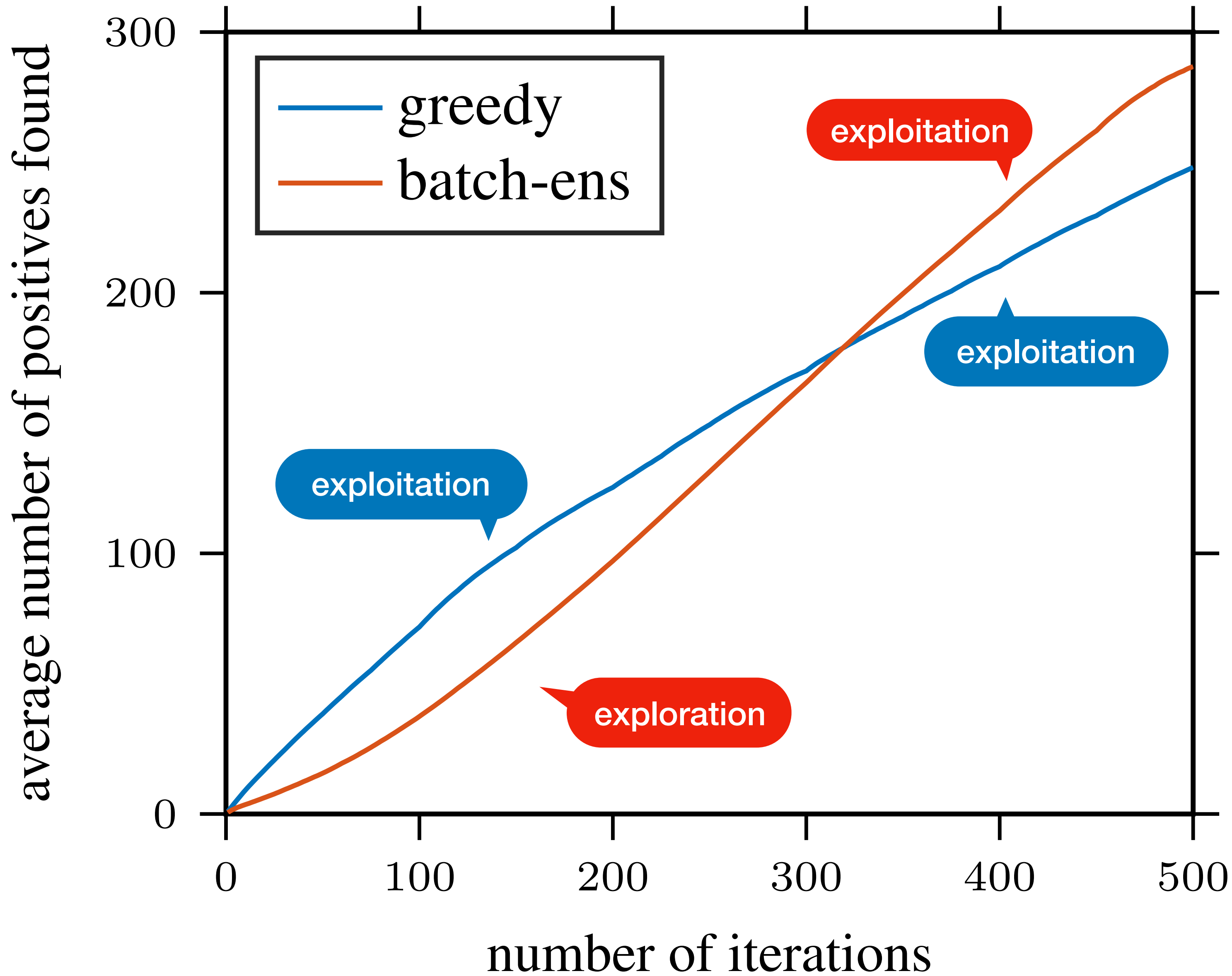
most likely label?

always positive?

always negative?

Best performing due to encouraging diversity

Empirical results



Averaged over 1600 experiments (10 drug discovery datasets, 8 batch sizes, and 20 repetitions each)

Batching is more efficient, but what are we compromising?

Batching is more efficient, but what are we compromising?

T=20

Batching is more efficient, but what are we compromising?

T=20

every point is chosen after observing the outcomes of all previous points!

(1 point / iter) * (20 iters) b=1

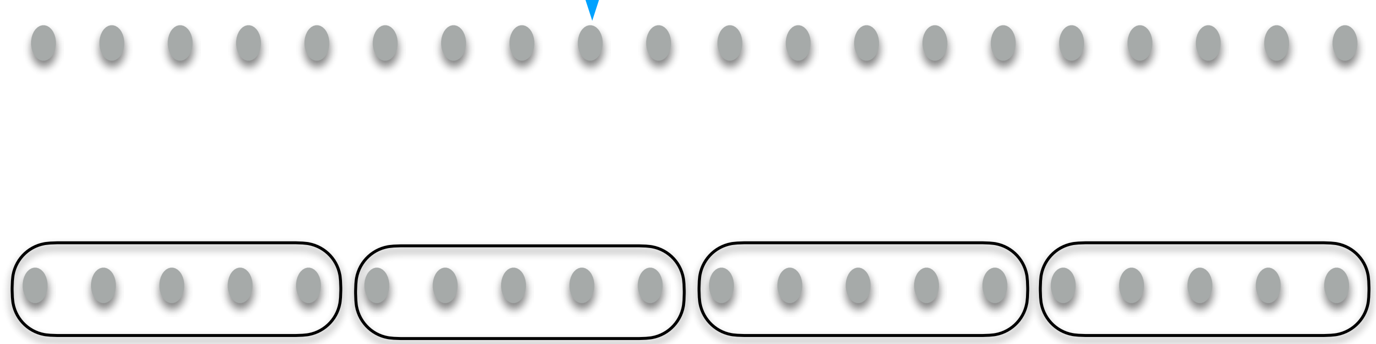
Batching is more efficient, but what are we compromising?

T=20

every point is chosen after observing the outcomes of all previous points!

(1 point / iter) * (20 iters) b=1

(5 points / iter) * (4 iters) b=5



points are chosen without observing the outcomes of previously added points in this batch

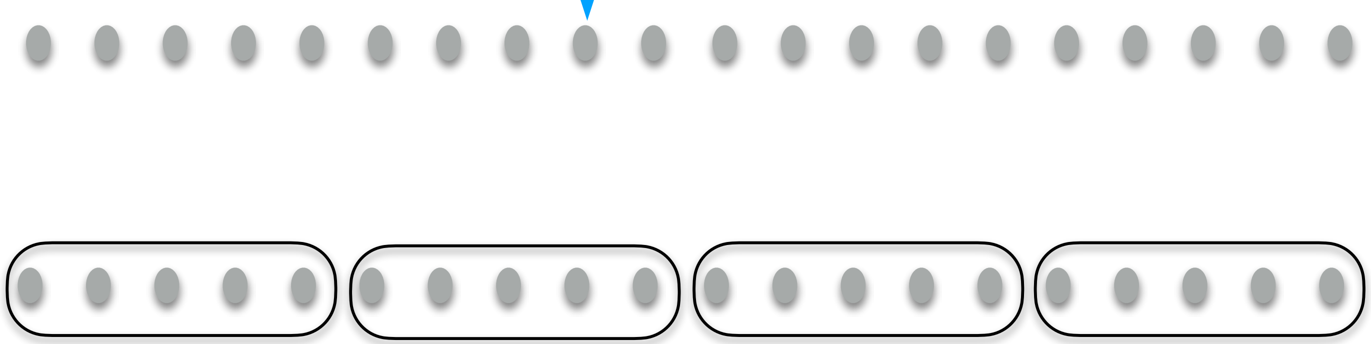
Batching is more efficient, but what are we compromising?

T=20

every point is chosen after observing the outcomes of all previous points!

(1 point / iter) * (20 iters) *b*=1

(5 points / iter) * (4 iters) *b*=5



points are chosen without observing the outcomes of previously added points in this batch

Less adaptive decisions could lead to worse performance!

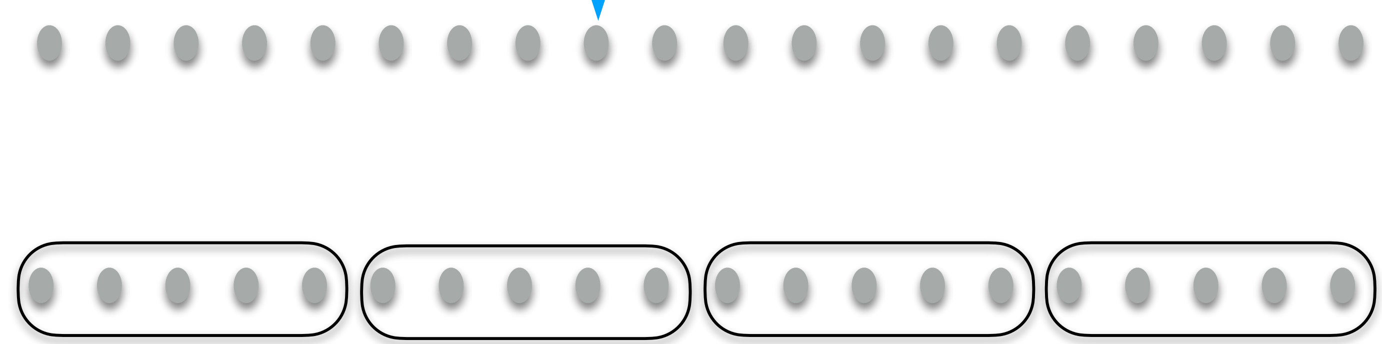
Batching is more efficient, but what are we compromising?

T=20

every point is chosen after observing the outcomes of all previous points!

(1 point / iter) * (20 iters) *b*=1

(5 points / iter) * (4 iters) *b*=5



points are chosen without observing the outcomes of previously added points in this batch

Less adaptive decisions could lead to worse performance!

But how much worse?

Adaptivity gap

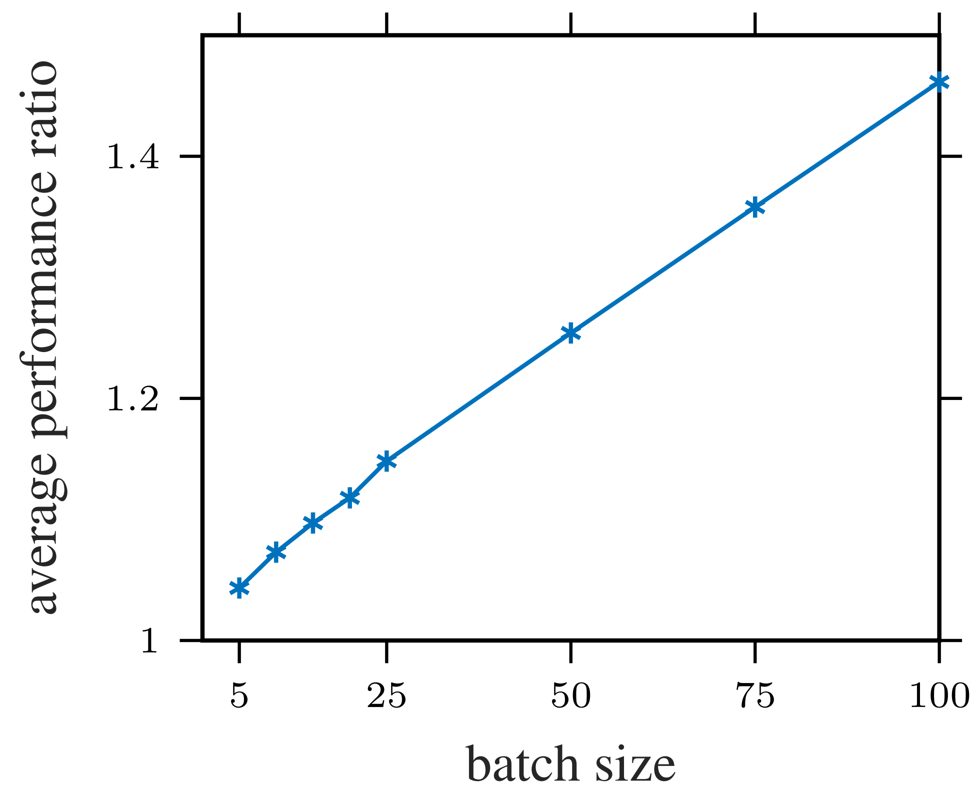
We prove that the performance ratio between optimal sequential and batch policies is at least linear in the batch size!

$$\frac{\mathrm{OPT}_1}{\mathrm{OPT}_b} = \Omega\left(\frac{b}{\log T}\right)$$

Adaptivity gap

We prove that the performance ratio between optimal sequential and batch policies is at least linear in the batch size!

$$\frac{\mathrm{OPT}_1}{\mathrm{OPT}_b} = \Omega\left(\frac{b}{\log T}\right)$$



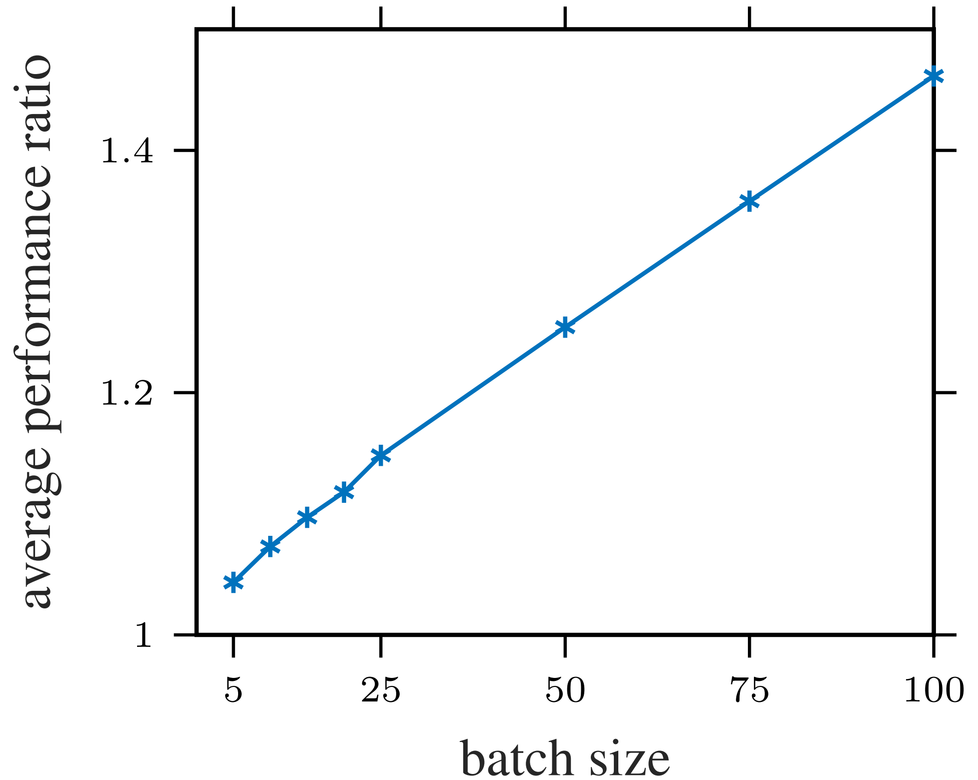
matching empirical results

Adaptivity gap

We prove that the performance ratio between optimal sequential and batch policies is at least linear in the batch size!

$$\frac{\mathrm{OPT}_1}{\mathrm{OPT}_b} = \Omega\left(\frac{b}{\log T}\right)$$

This insight could help us choose the batch size in cases where we have many options.



matching empirical results

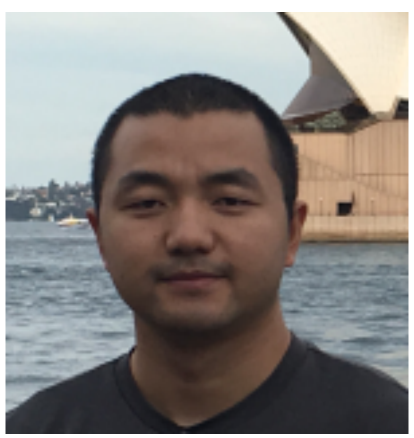
Washington University in St. Louis

CAPES

MISSOURI TRANSECT

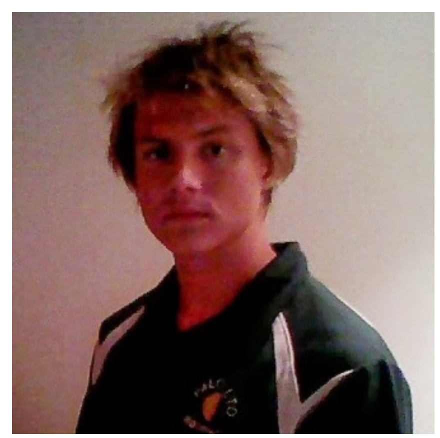
NSF

Thanks for your attention!
Poster: #131



Shali Jiang

Gustavo Malkomes



Matthew Abbott

Benjamin Moseley

Roman Garnett

NeurIPS 2018