Bayesian inference

Computational Cognitive Science 2014
Lecture 3
Dan Navarro

The lotto problem
(“this is computer science and not just maths, right?”)

Bizarro lotto inference game

- The Bizarro company runs a lotto.
  - Each day they announce a winning number, x
  - The winning number is an integer from 1 to 100
  - But, during any given week, the winning number is chosen at random from an unknown range between l and u.
  - In other words: $1 \leq l \leq x \leq u \leq 100$
  - At the end of the week, the numbers l and u are revealed, and new value chosen.

Bizarro lotto inference game

- An example...
  - On Sunday, the company chooses $l = 15$, $u = 39$.
  - But they don't tell these numbers to anyone.
  - They then run the lotto during the week…

    Mon: 31,

    Tue: 15,

    Wed: 37,

    Thu: 20,

    Fri: 20
  - On Saturday, the company reveals l and u

The bookie's problem

- A friend of mine wants to offer side bets.
  - Anyone can select a number y on any day of the week, and if y is between l and u, they win
  - If he wants all possible bets to be fair, what odds should she offer for y?

- Can we build a model to solve this?

What does the bookie need to know?

- Let $X = (x_1, \ldots, x_k)$ be the lotto data for k days
- That is x_i is the winning number on day i
- Let $C = (l, u)$ be the true range
- Our bookie needs to know the probability that y is in C, given that we've seen data X so far,

$$P(y \in C | X)$$

Sample space and hypothesis space

- Sample space
  - The lotto numbers are between 1 and 100
  - Sample space X is the set (1, 2, 3, … , 100).

- Hypothesis space
  - Each hypothesis h specifies a possible choice of integers l and u, such that $1 \leq l \leq u \leq 100$
  - So H is the set of all such choices
  - There's 5050 of these! Time for some coding…

Specify the prior distribution

- The company chooses the true values at random, so P(h) is uniform across the 5050 hypotheses

$$P(h) \propto \frac{1}{|H|} = \frac{1}{5050}$$

The likelihood

- Each winning number x is selected uniformly at random from the range (l, u)

- Notation:
  - Let $|h| = u - l + 1$ be the size of h
  - and $x \in h$ means $l \leq x \leq u$

- Likelihood for a single observation:

$$P(x|h) = \begin{cases} \frac{1}{|h|} & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases}$$

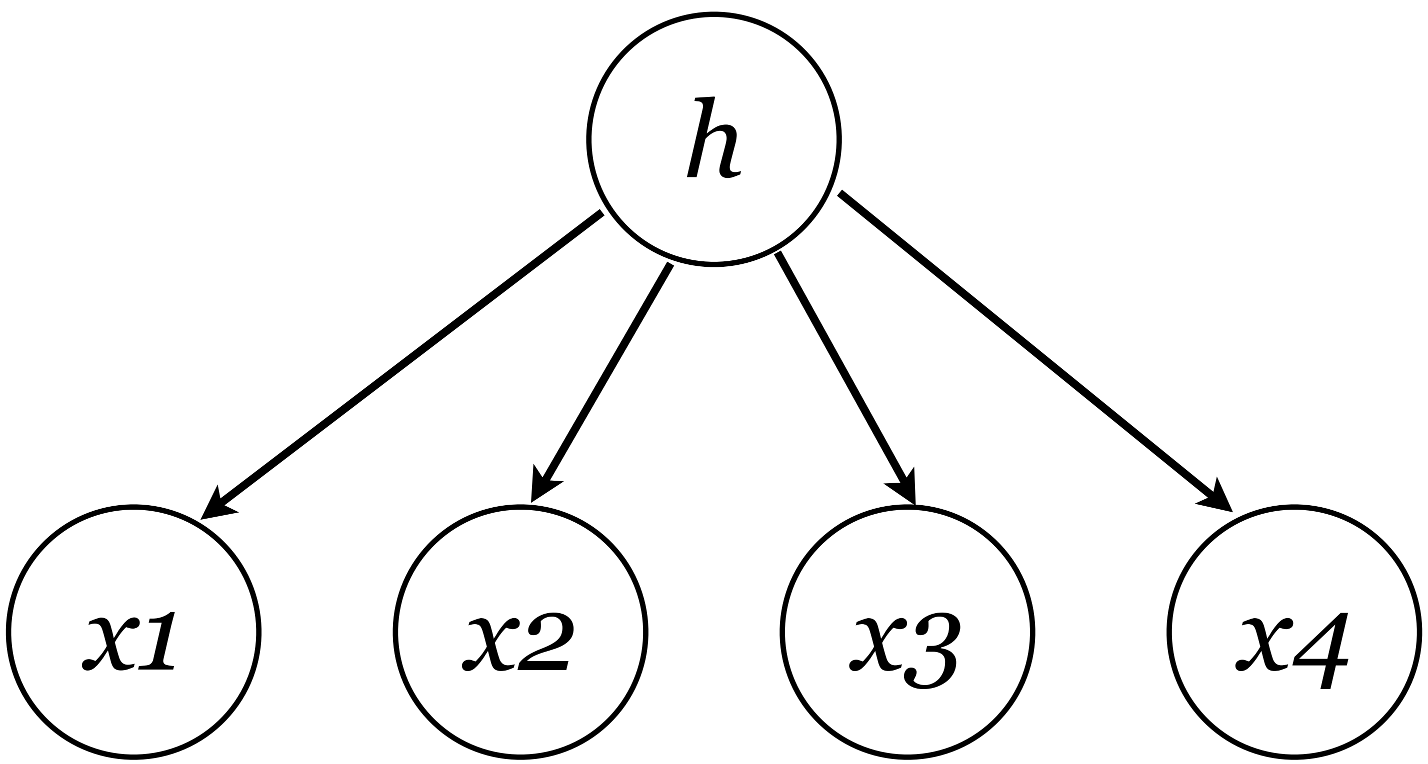
The likelihood for multiple observations

- The lotto numbers are independently drawn from the range between l and u
- If h is the correct hypothesis about the range, then we can just multiply the individual probabilities...

$$\begin{aligned} P(X|h) &= P(x_1, x_2, \ldots x_k|h) \\ &= \prod_{i=1}^{k} P(x_i|h) \end{aligned}$$

The likelihood for multiple observations

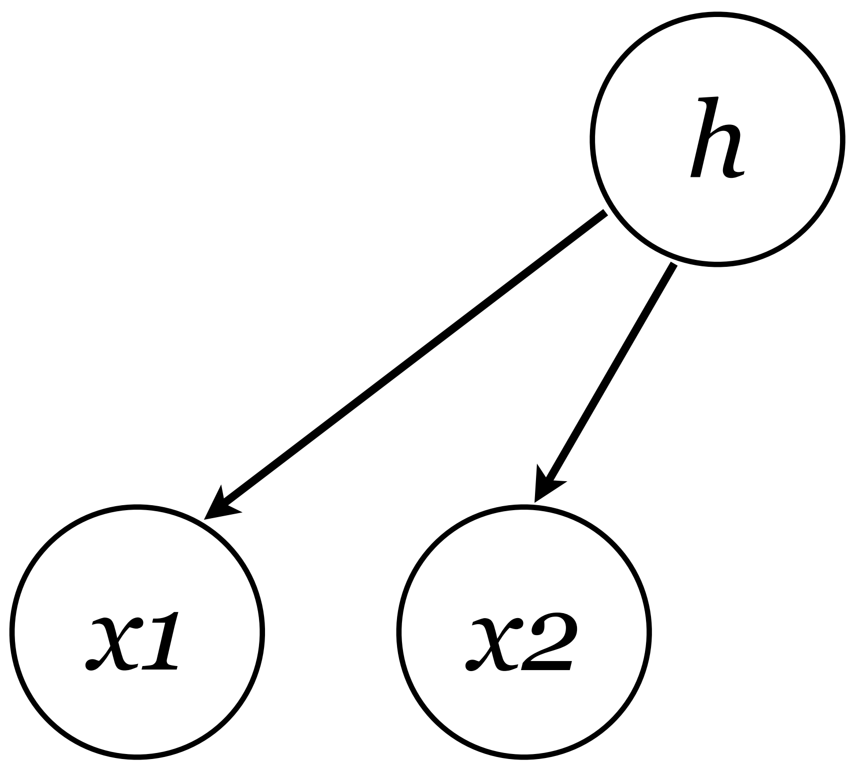
- It's important to understand what's happening here
- Here's a graphical illustration:



All of the winning numbers (x) are "generated" from the true hypothesis h

The likelihood for multiple observations

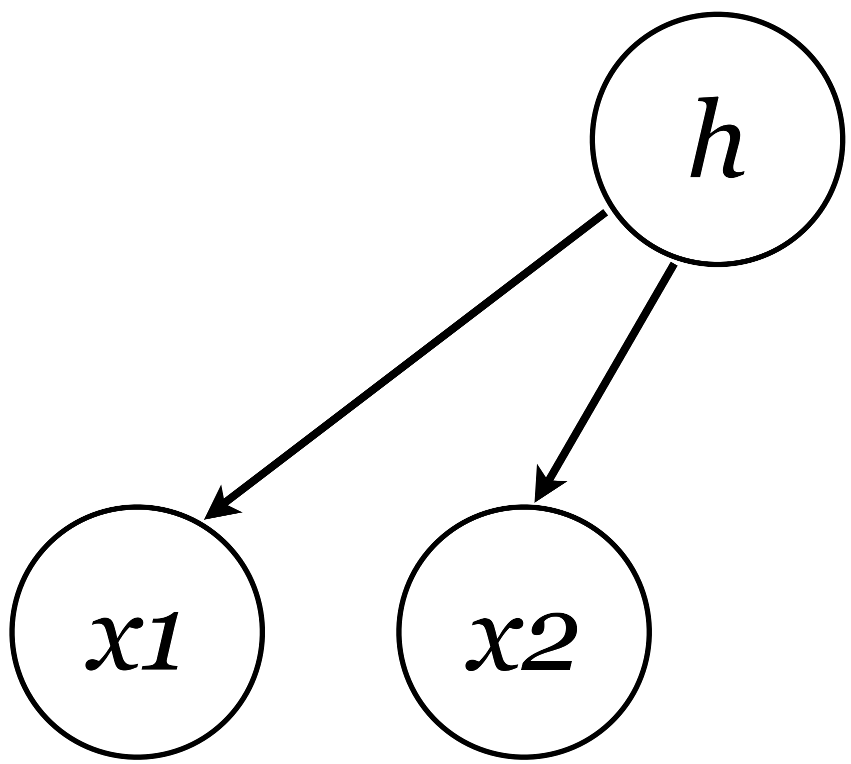
- It's important to understand what's happening here
- Here's a graphical illustration:



Everything you need to know about the probability of x1 value is captured by h ... i.e., if you know h, then x2 tells you nothing new about x1

The likelihood for multiple observations

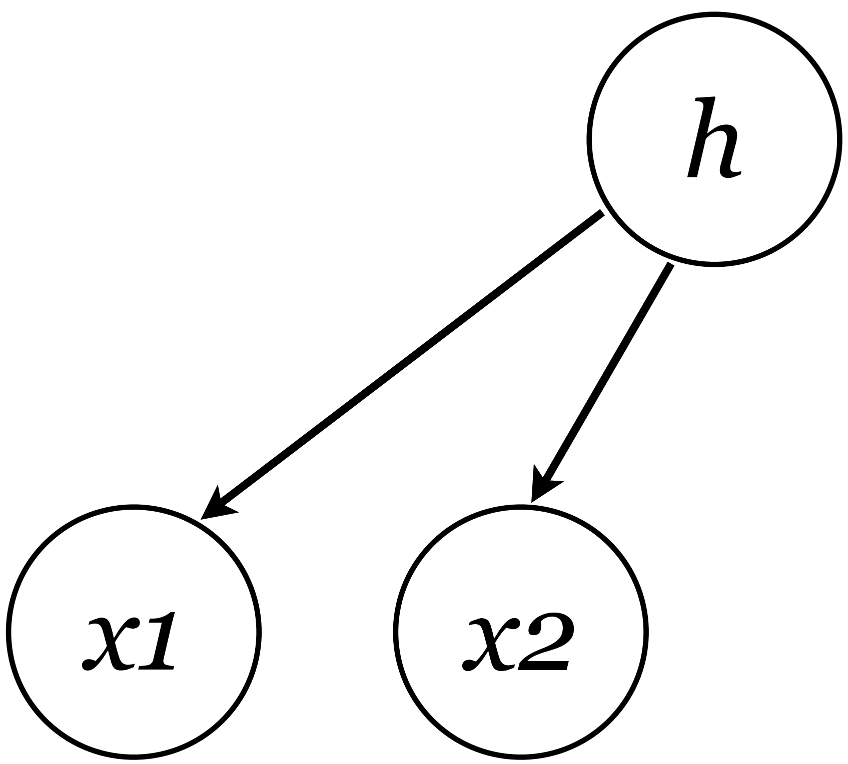
- It's important to understand what's happening here
- Here's a graphical illustration:



We say that x2 and x1 are conditionally independent given h

The likelihood for multiple observations

- It's important to understand what's happening here
- Here's a graphical illustration:

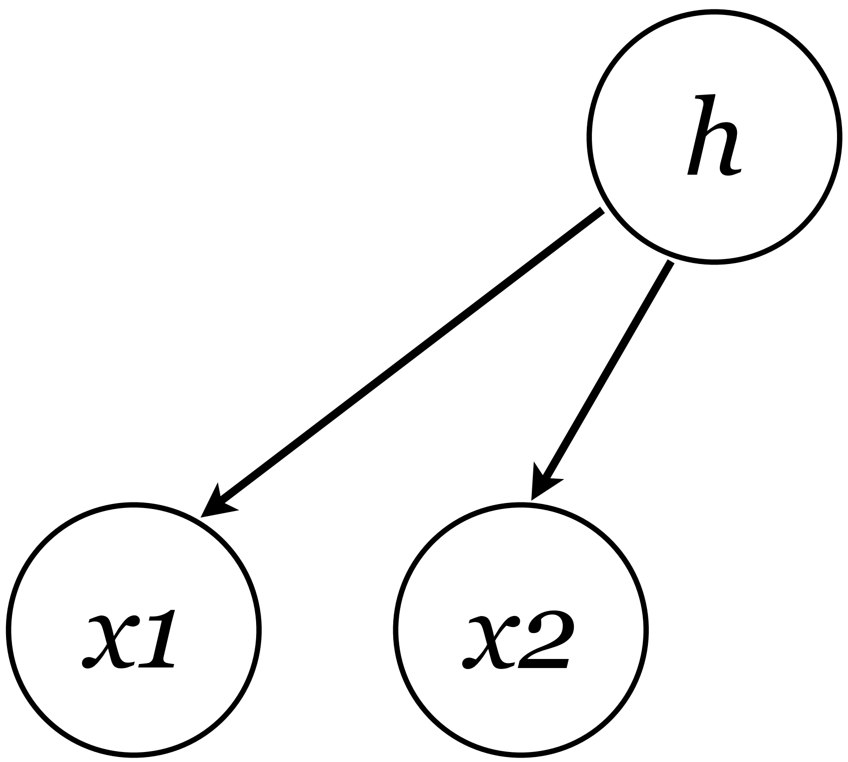


Mathematically, this means that the likelihood function factorises as follows:

$$P(x1, x2 \mid h) = P(x1 \mid h)\, P(x2 \mid h)$$

The likelihood for multiple observations

- It's important to understand what's happening here
- Here's a graphical illustration:



In our example, the multiplication is really, really simple:

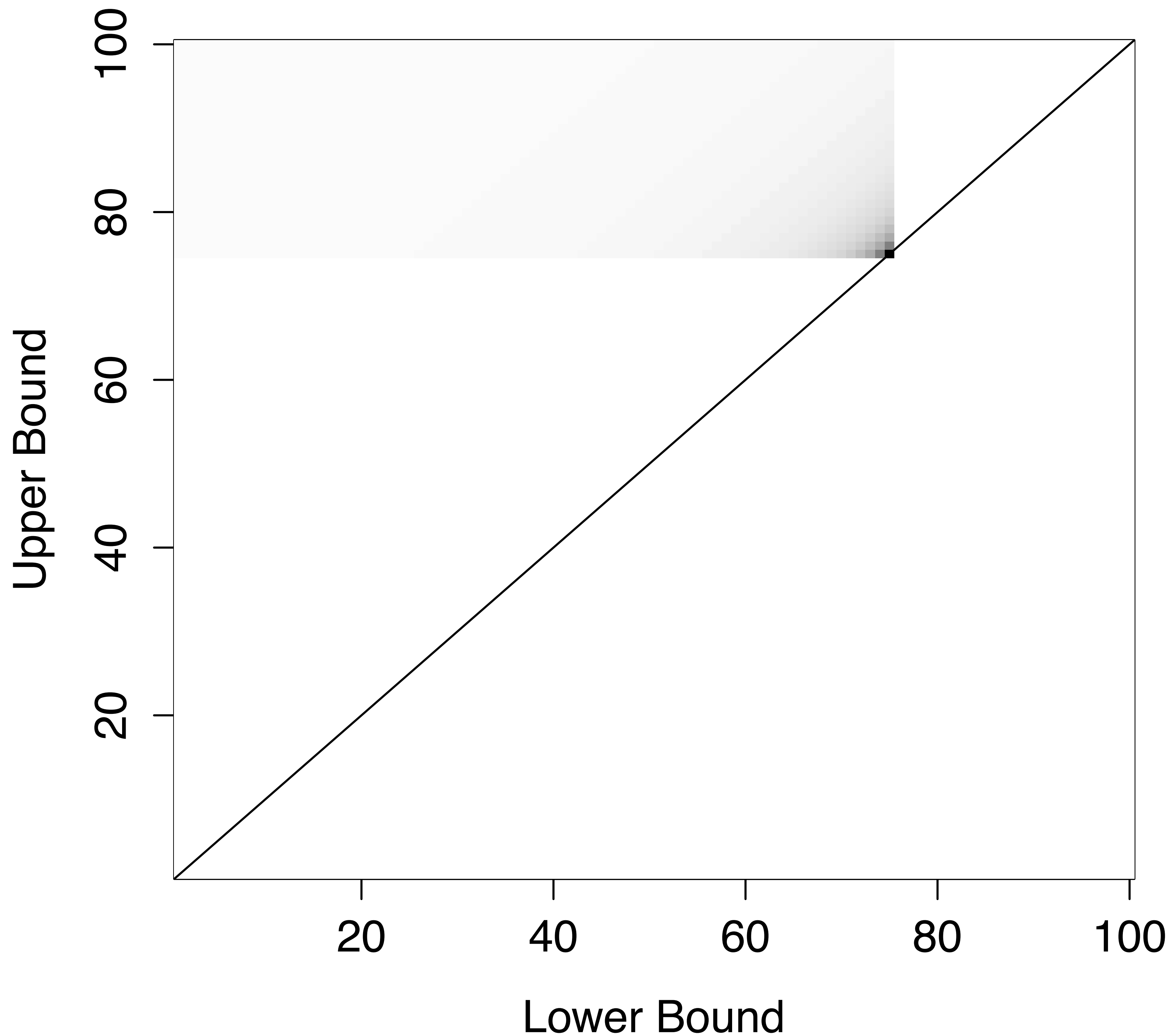
$$P(x1, x2 \mid h) = P(x1 \mid h)\, P(x2 \mid h)$$

$$= (\, 1 \,/\, |h| \,)\, (\, 1 \,/\, |h| \,)$$

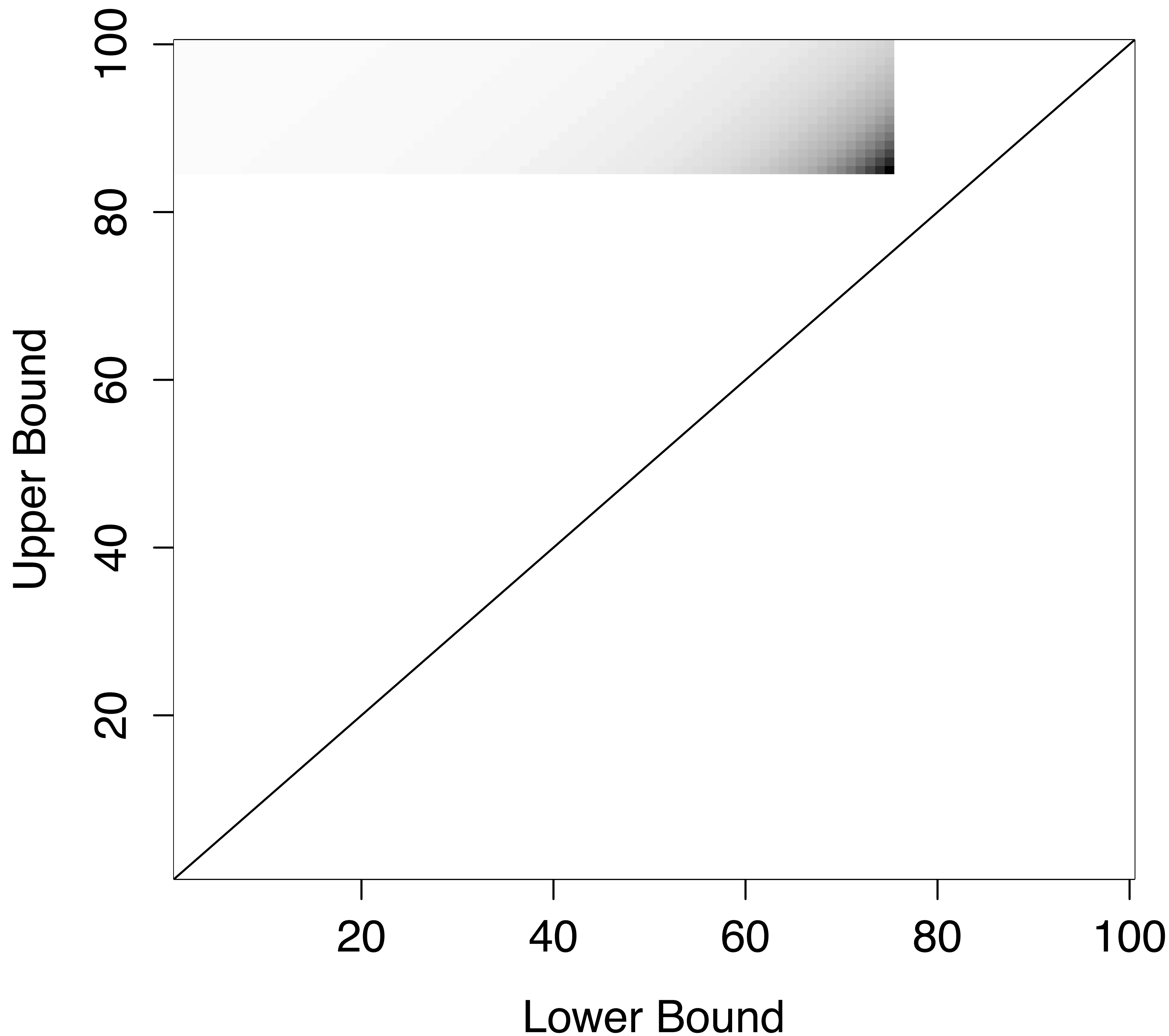
We can now solve our inference problem

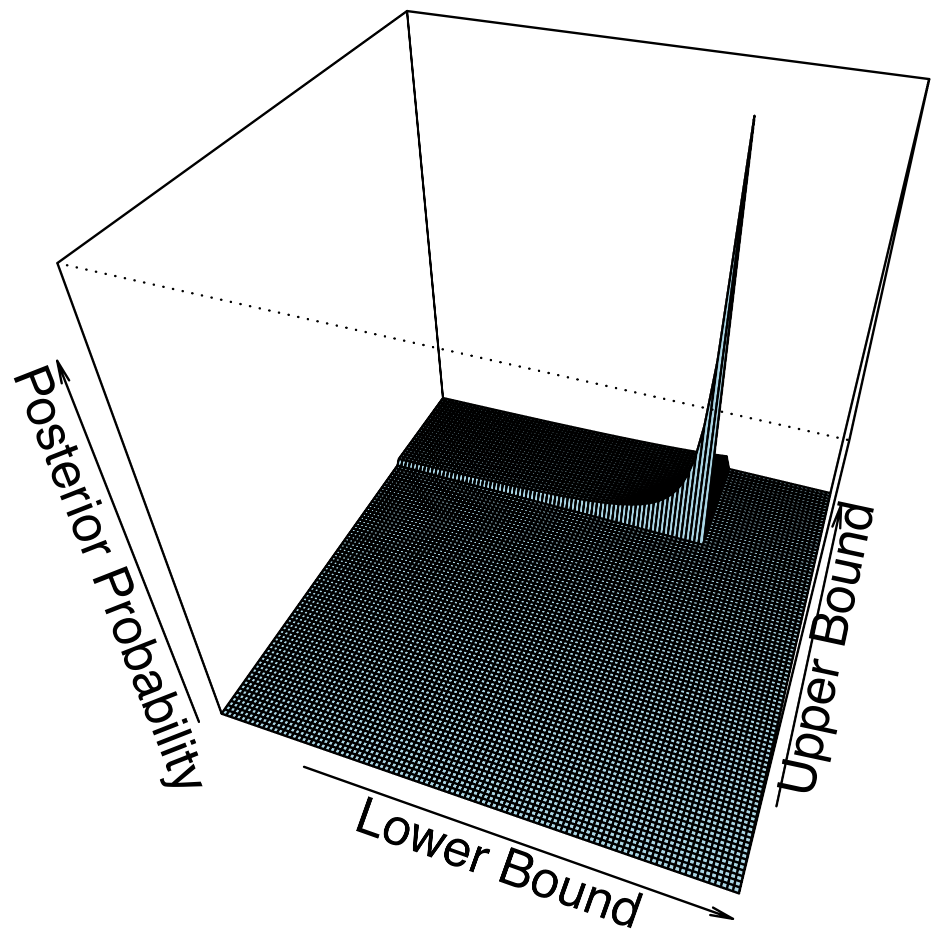
$$P(h|X) = \frac{P(X|h)P(h)}{\sum_{h' \in \mathcal{H}} P(X|h')P(h')}$$

posterior after one observation at 75

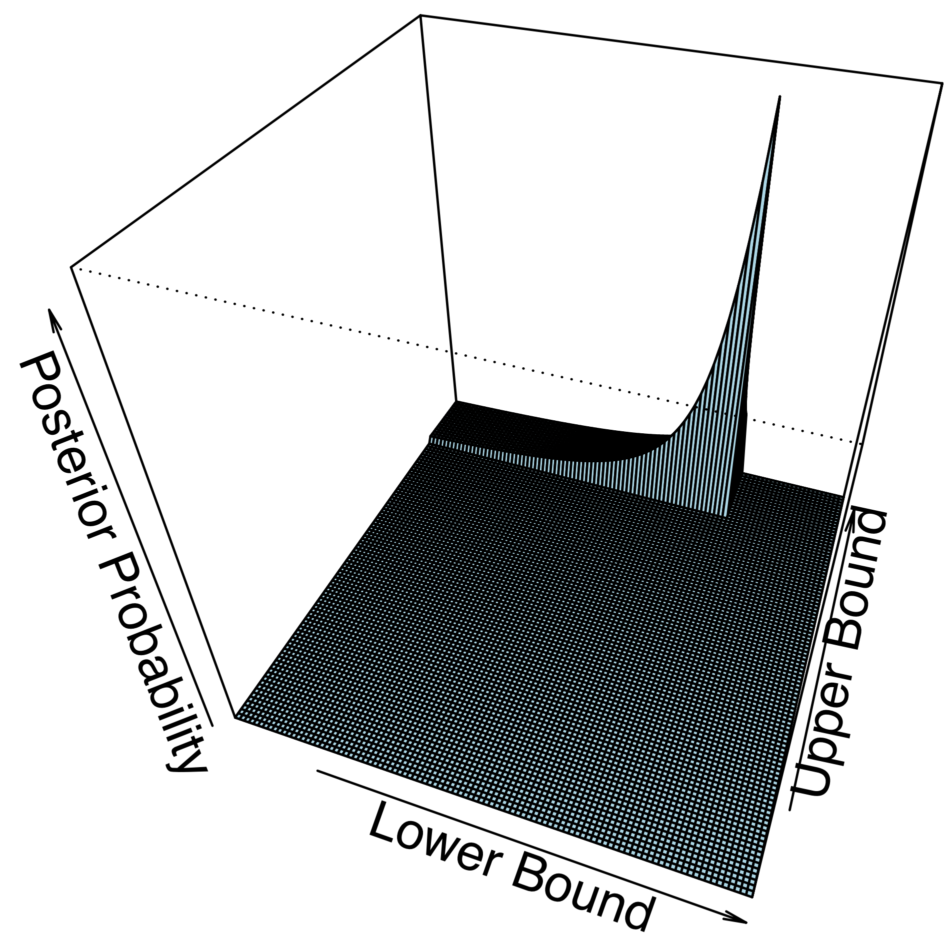


posterior after two observations at 75 & 85





posterior after one observation at 75



posterior after two observations at 75 & 85

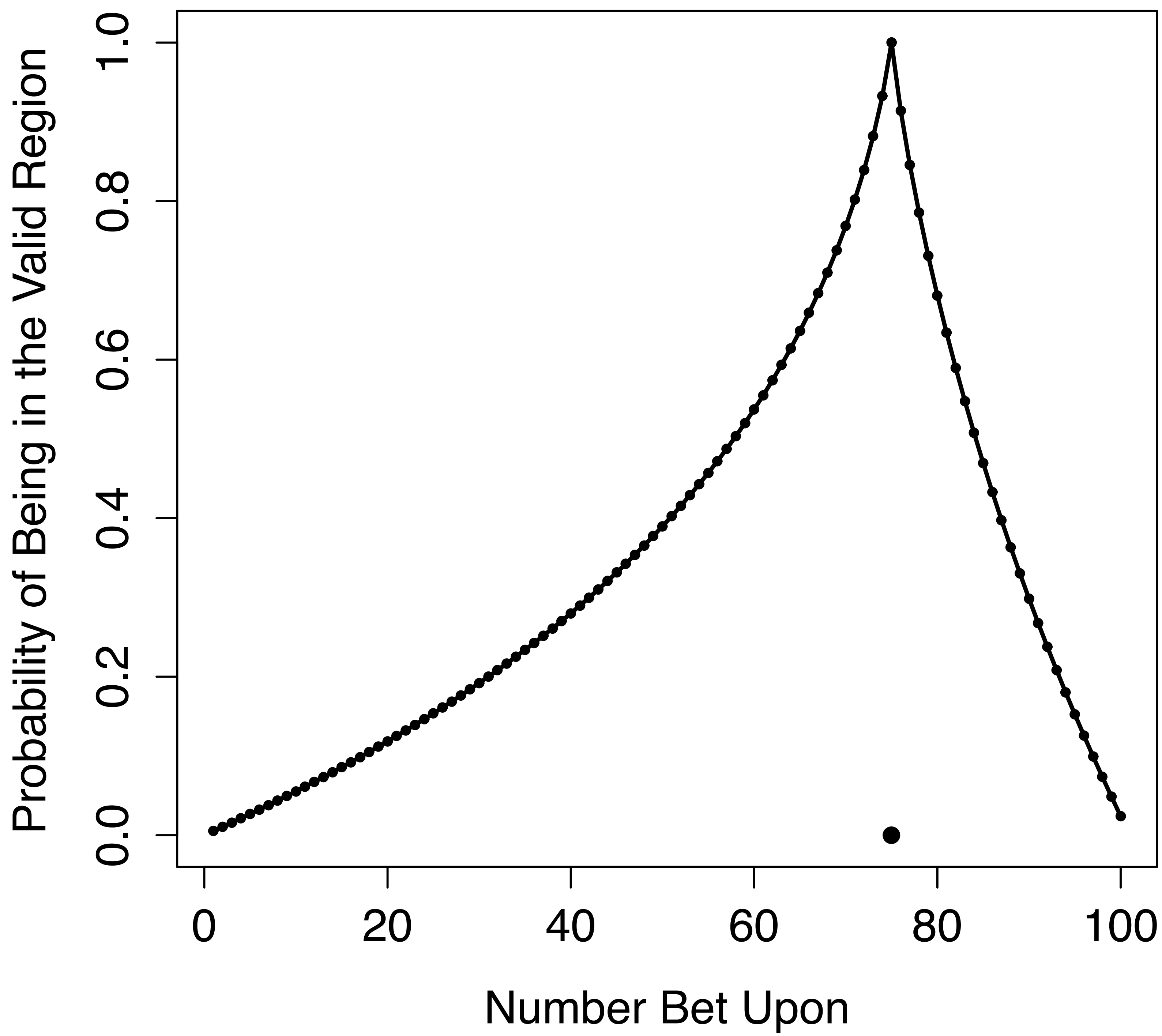
Answering the bookie's question

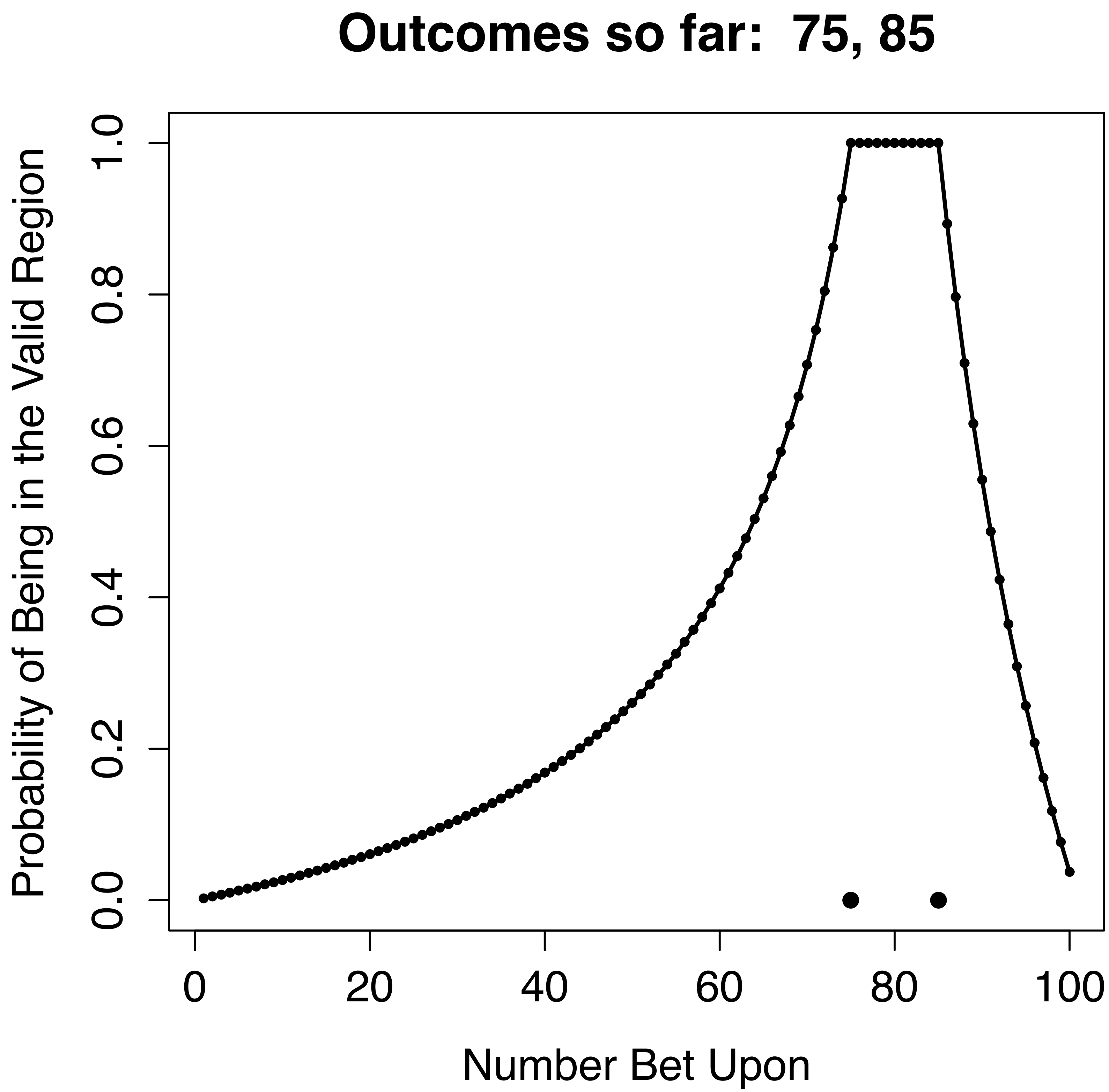
- To calculate the probability that y falls within the true range C

$$P(y \in C|X) = \sum_{h \in \mathcal{H}} P(y \in C|h)P(h|X)$$

- where $P(y \in C|h)$ equals 1 if y is within h, and equals 0 if it doesn't

Outcomes so far: 75





Outcomes so far: 75, 85
Probability of Being in the Valid Region
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Number Bet Upon

Demonstration: lotto.R

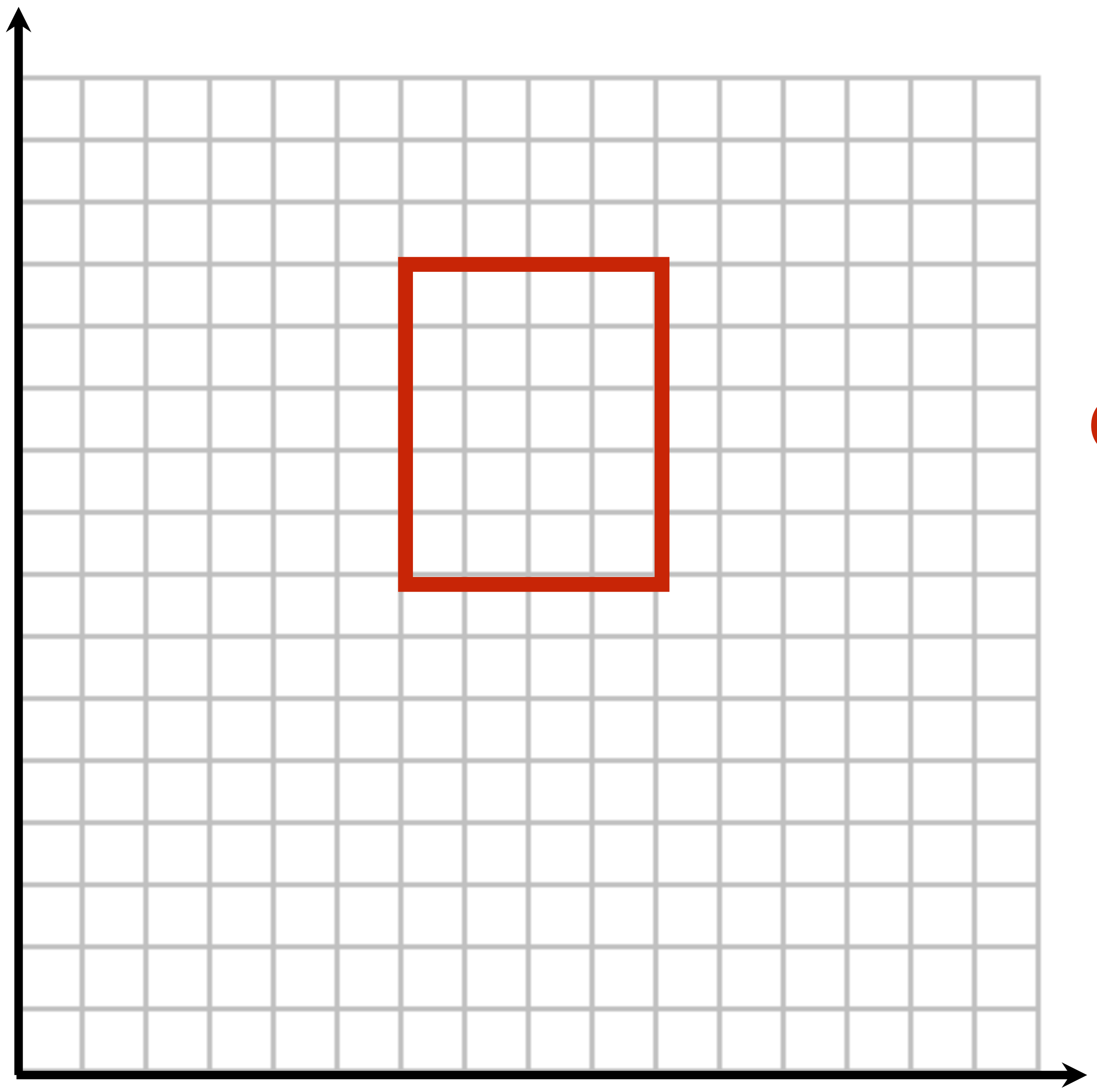
(FYI: the lotto problem is formally equivalent to an interesting psychological problem that Amy will talk about later)

Winning at battleships
(“Ockham’s razor”)

Ockham's razor

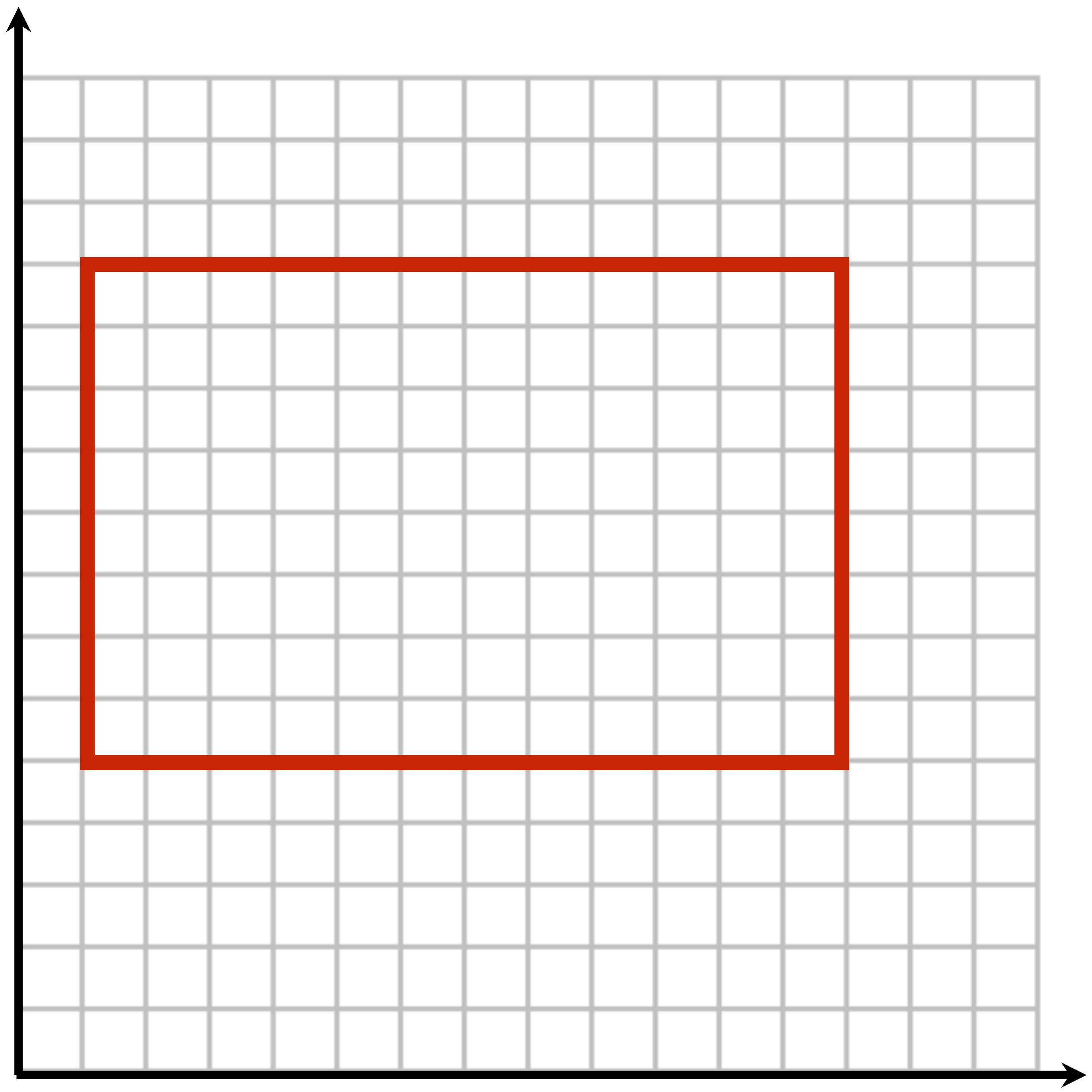
- What is it?
  - "Do not multiply entities beyond necessity"
  - The "simplest" explanation that "fits the data" is most likely to be correct
- How do we formalise it?
  - We need to understand what we mean by simplicity
  - And we need some rule that favours it
- Formalising simplicity is hard!
  - I'll show you the easy way, and (maybe) talk in passing about the hard way…

Generalised battleships!



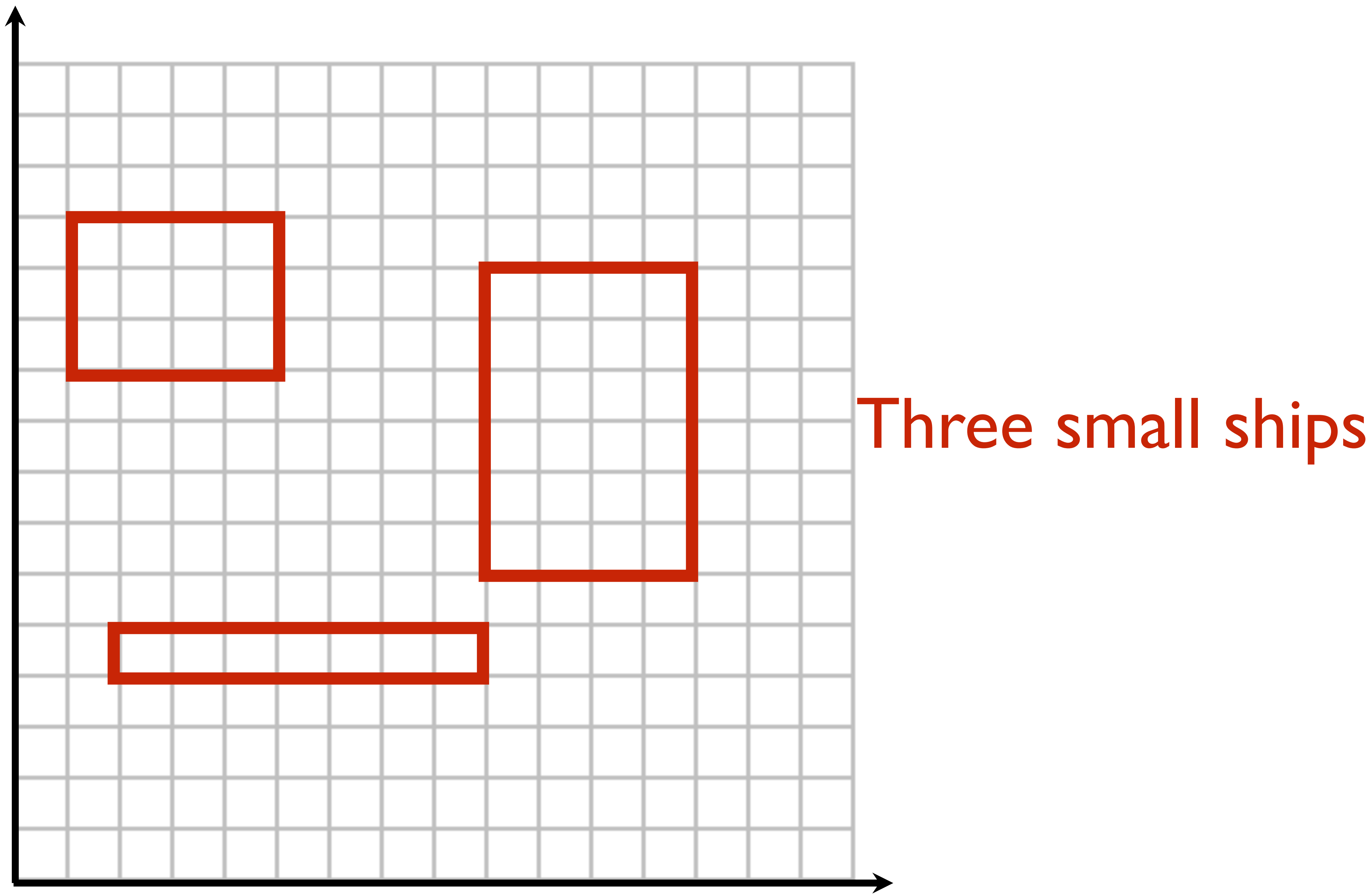
One small ship

Generalised battleships!

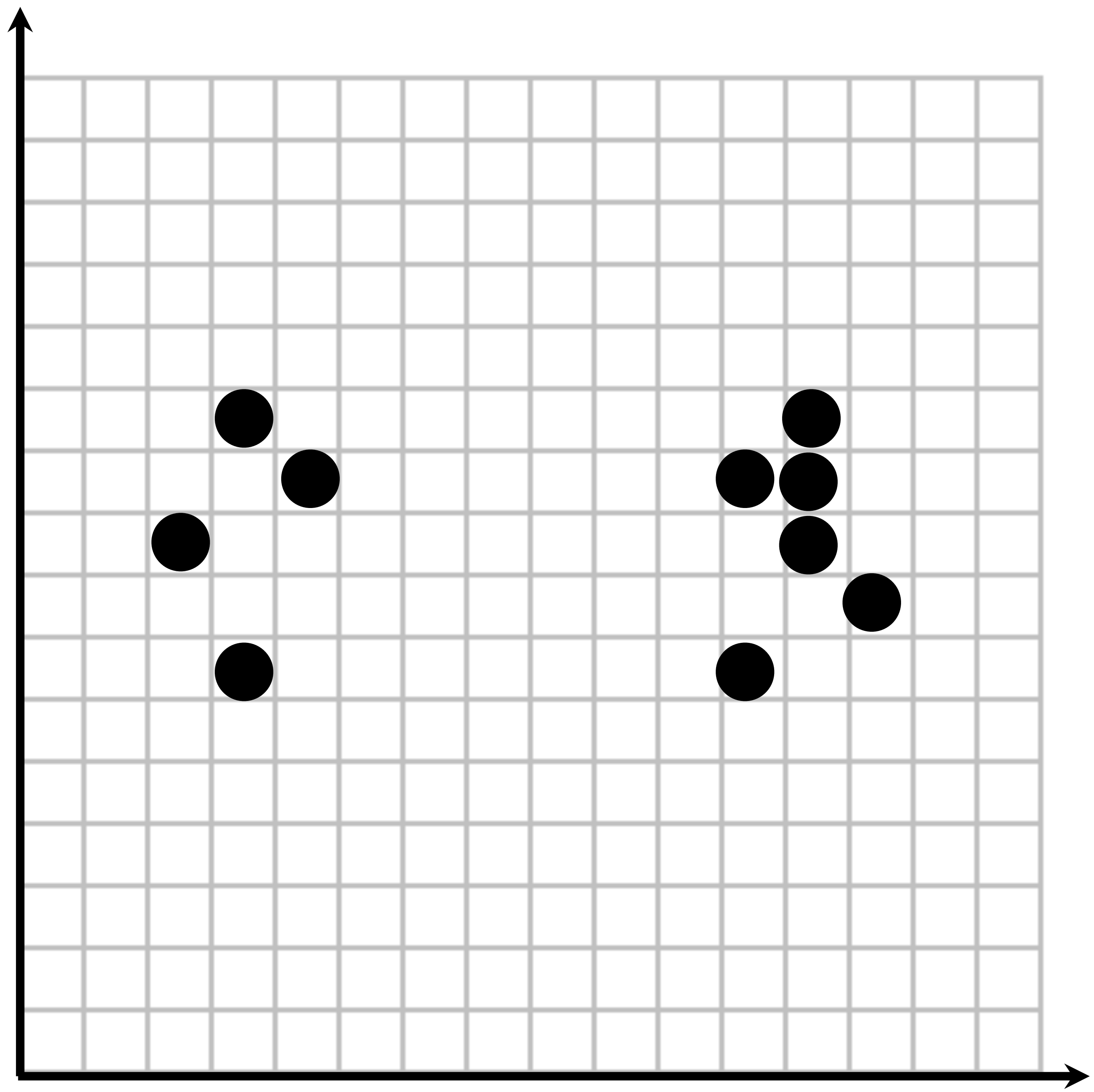


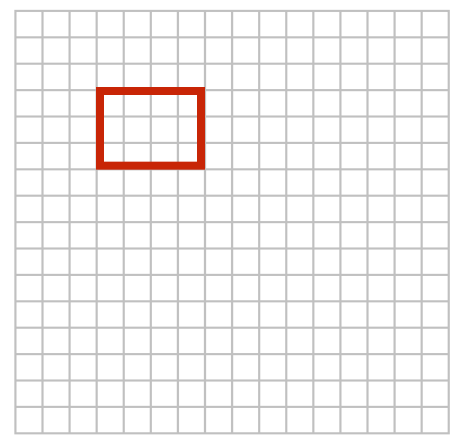
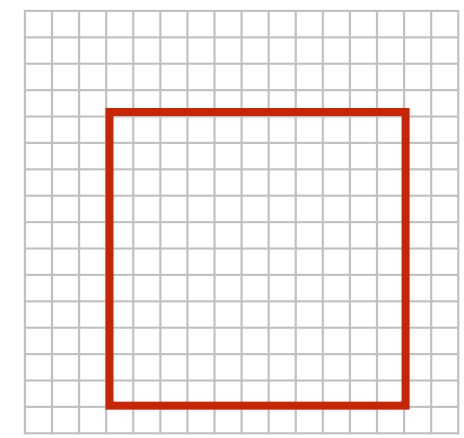
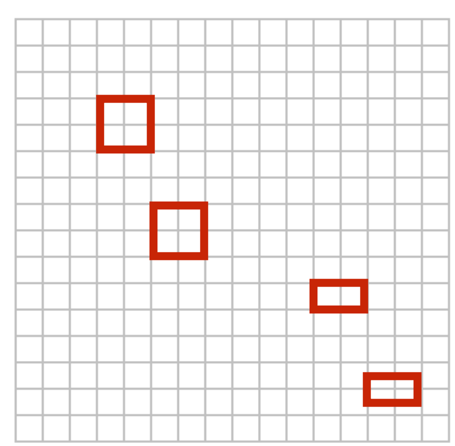
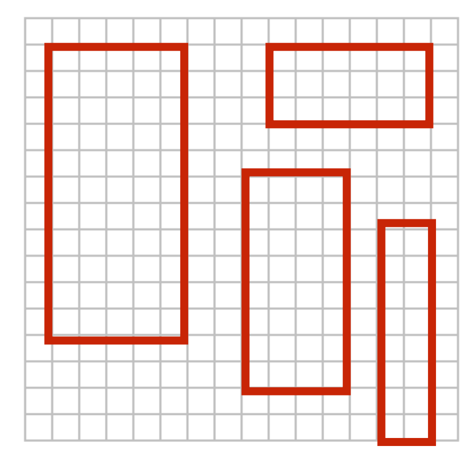
One large ship

Generalised battleships!



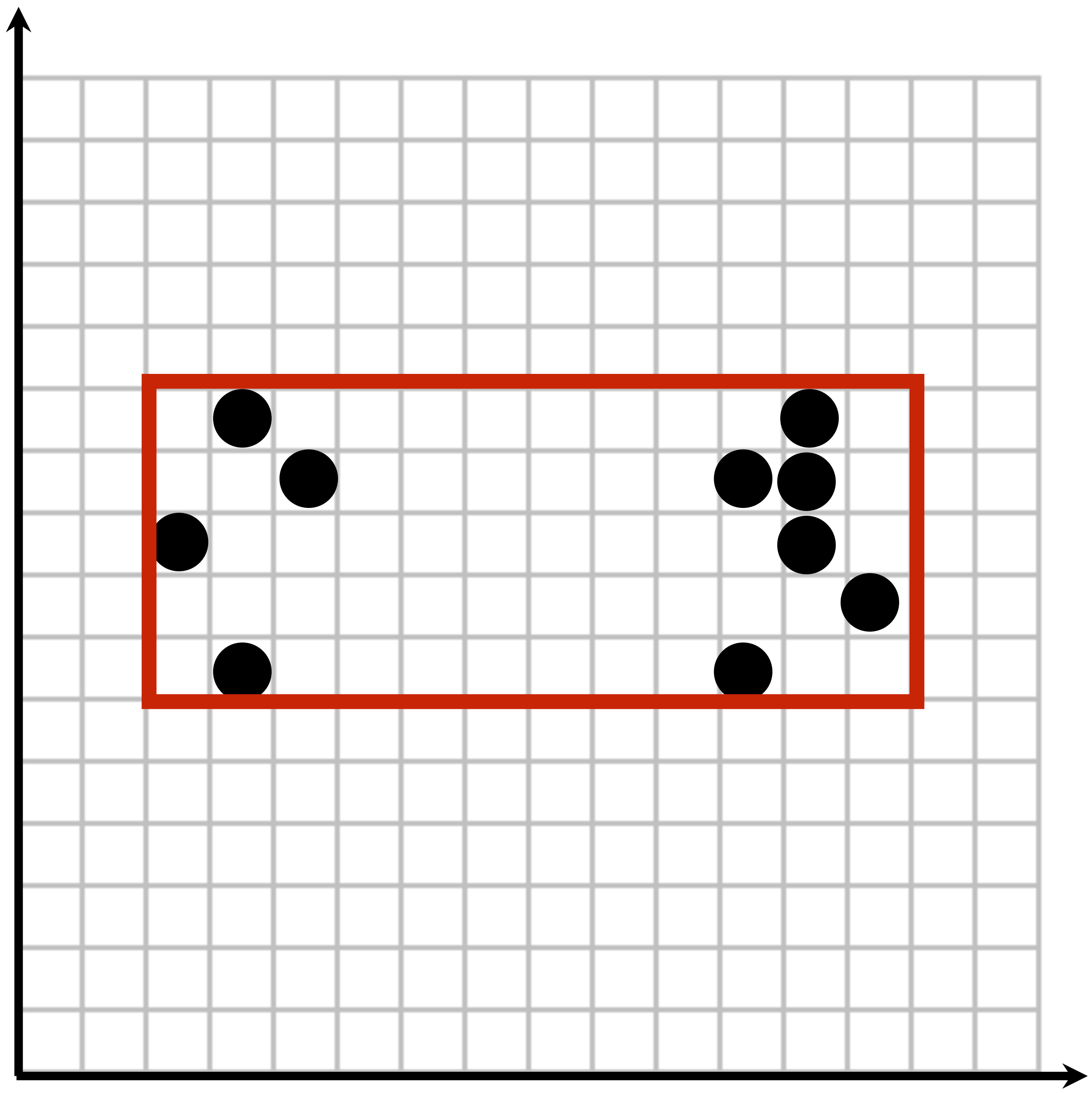
On each turn, you get to see a randomly sampled “hit”



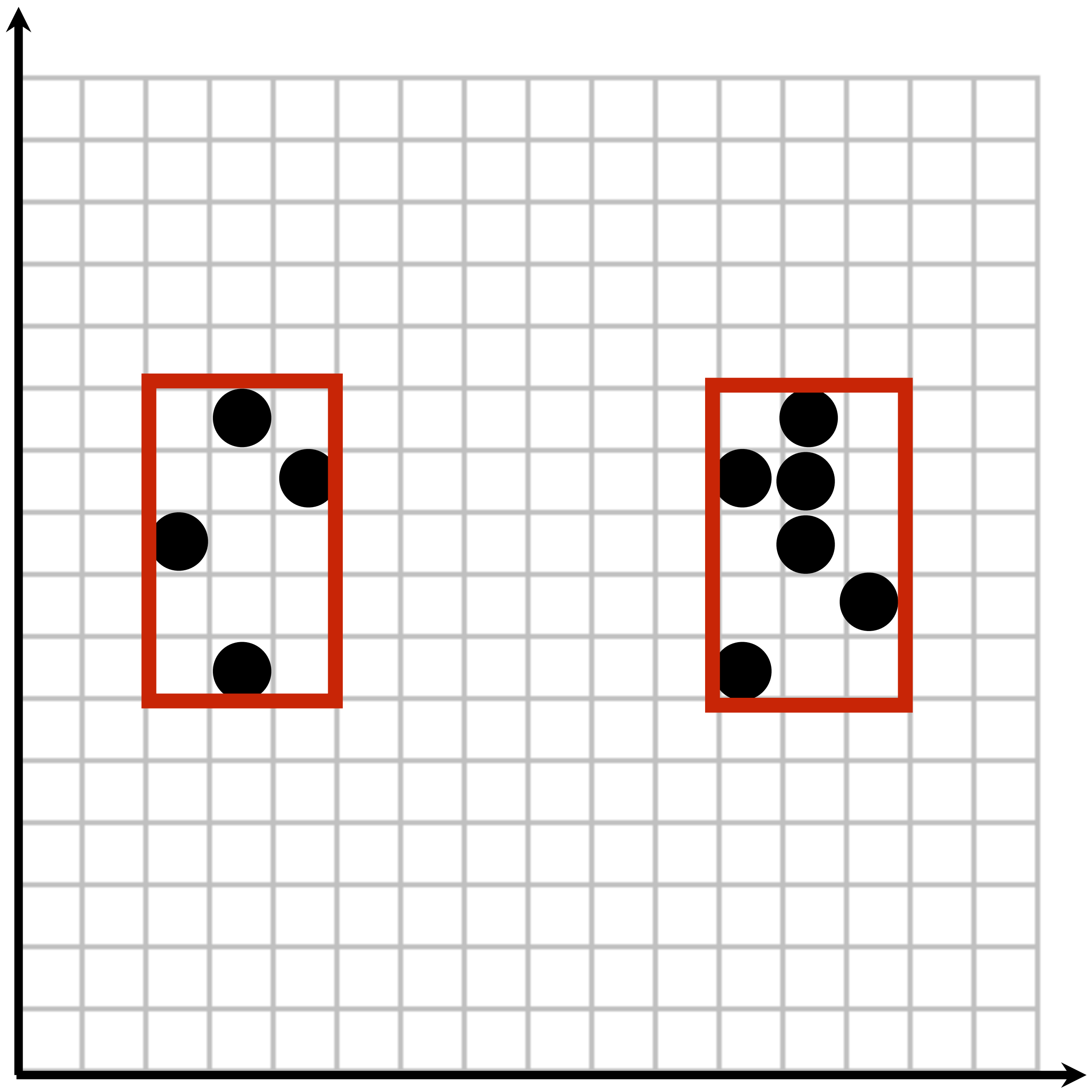
	consistent with very few possible observations (12 squares covered)	consistent with many possible observations (121 squares covered)
consists of few distinct “entities” (1 ship)		
consists of many distinct “entities” (4 ships)		

Which of the following is the “simplest explanation” that is “consistent with data?”

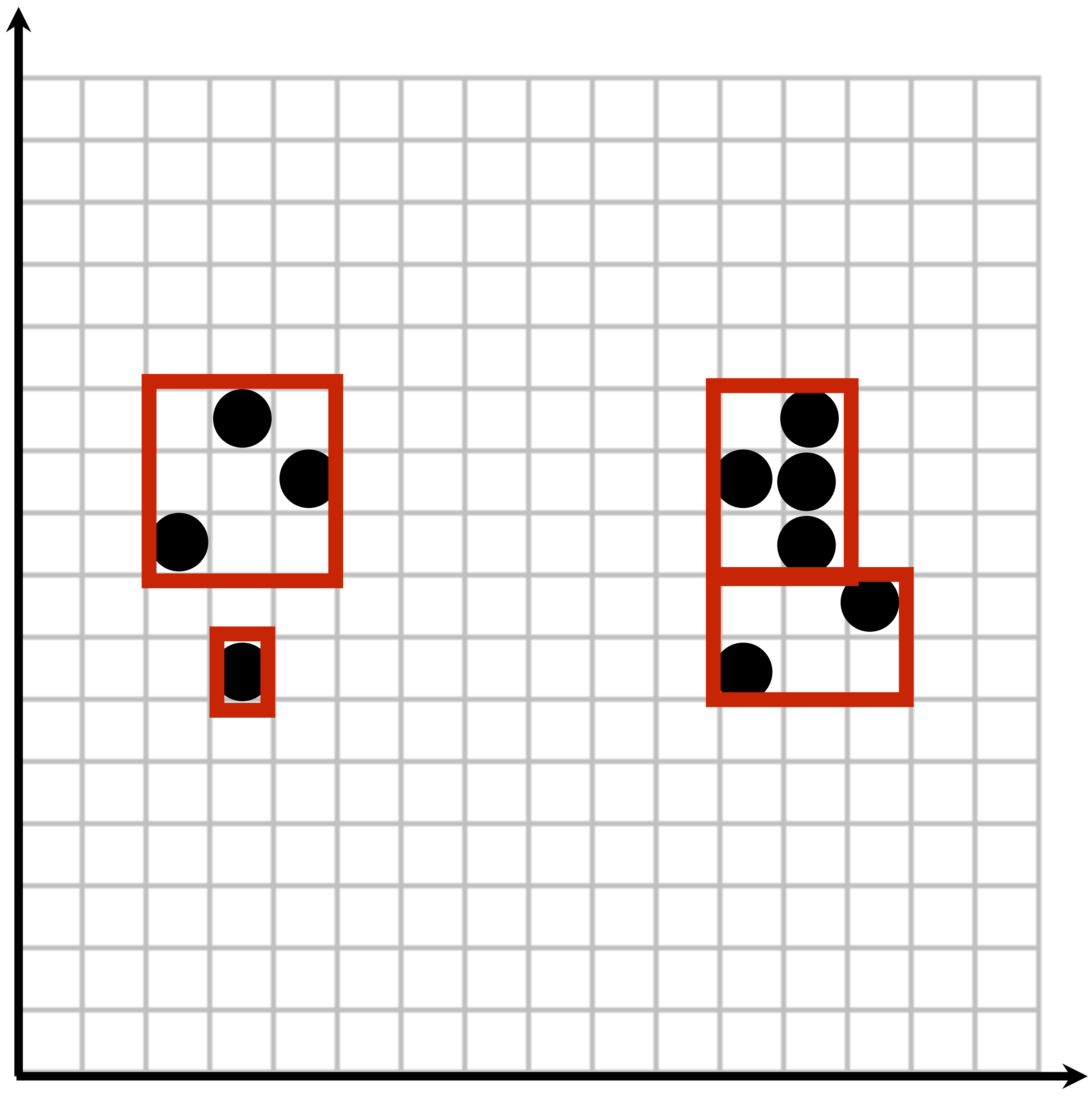
1 entity, 60 squares covered



2 entities, 30 squares covered



4 entities, 22 squares covered

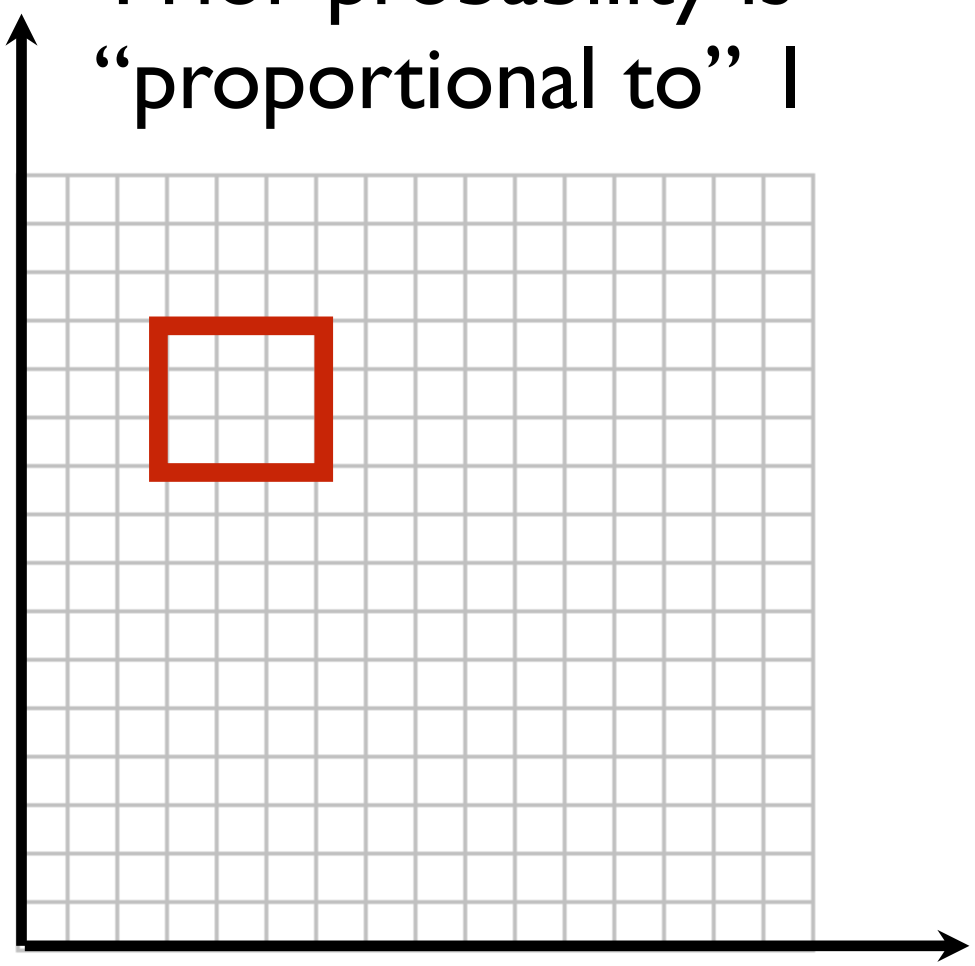


Simplicity: the Bayesian view

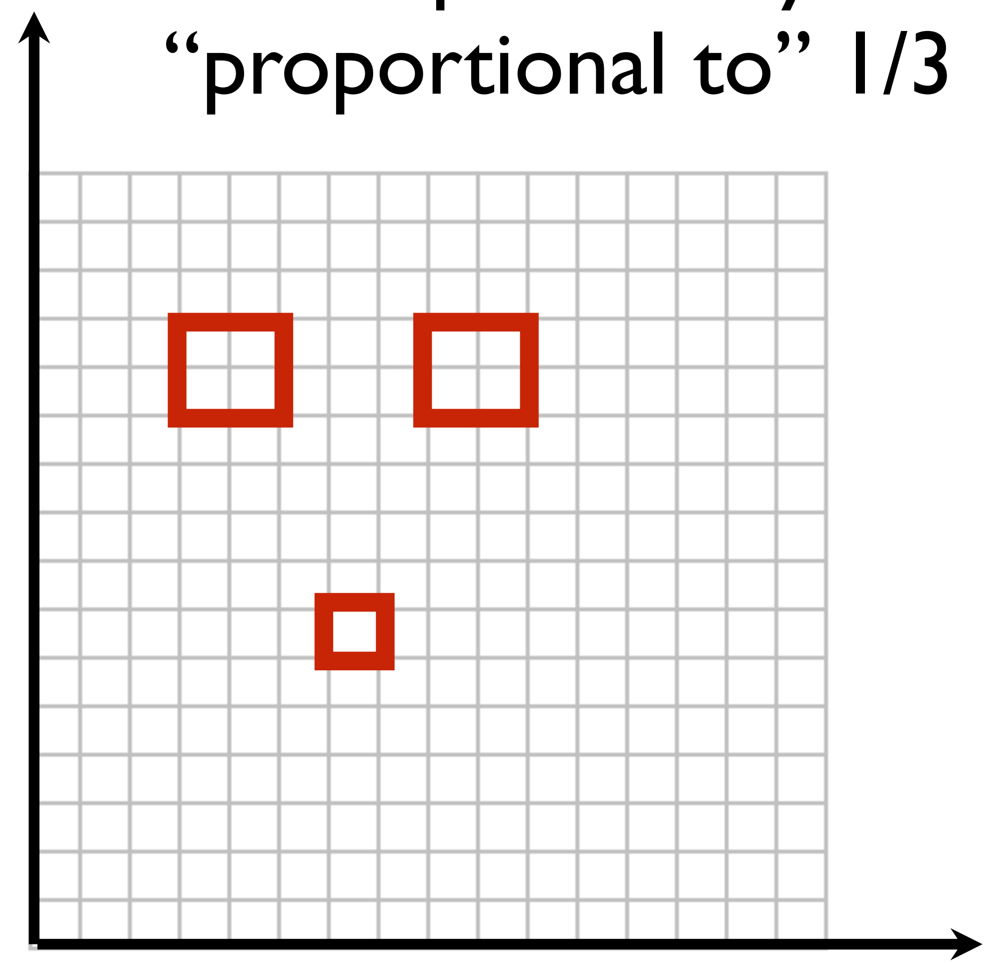
$$P(h) \propto \frac{1}{N_e}$$

Choose a **prior** to favour simplicity: prior probability decreases as a function of the number of entities

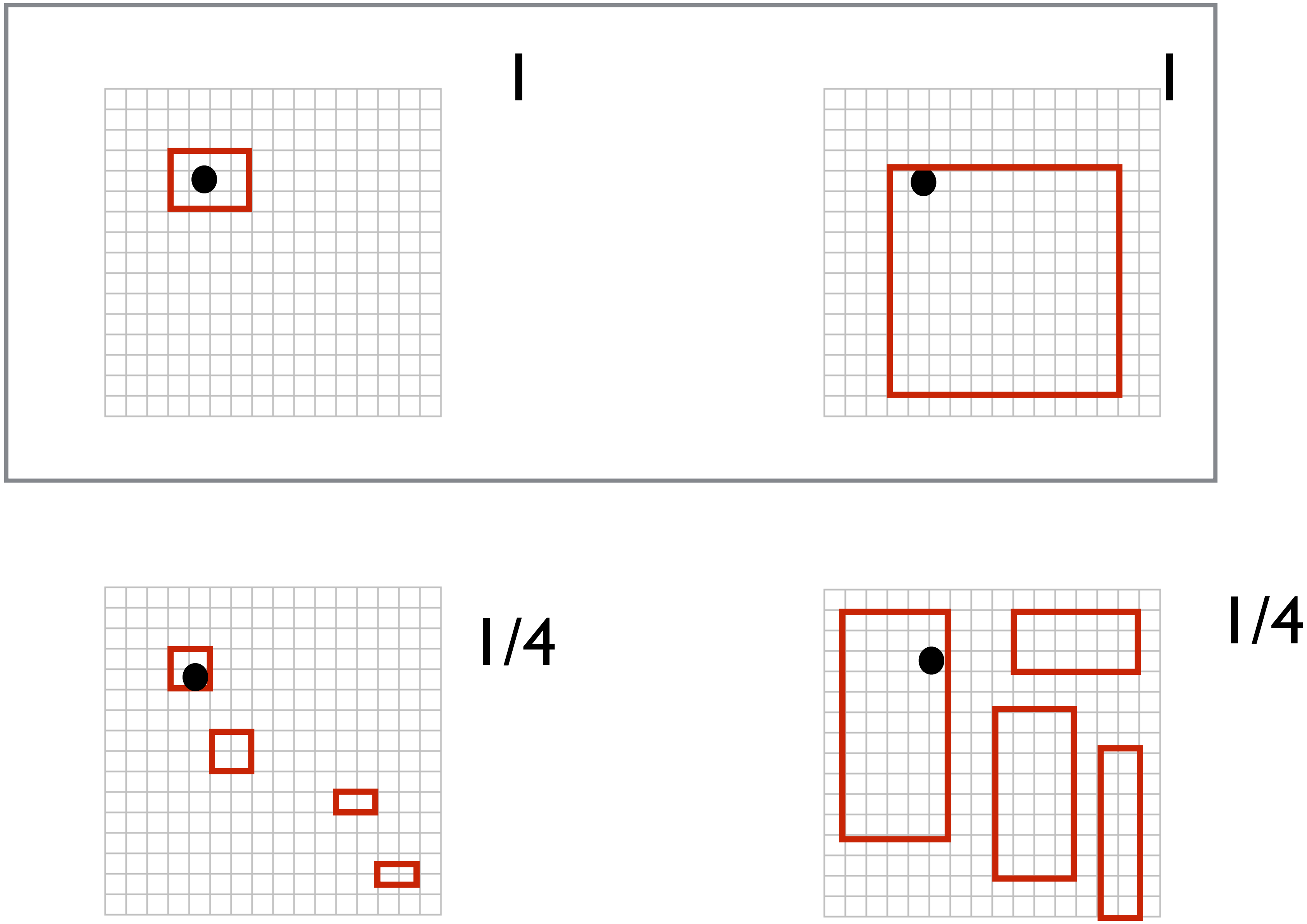
Prior probability is "proportional to" 1



Prior probability is "proportional to" 1/3



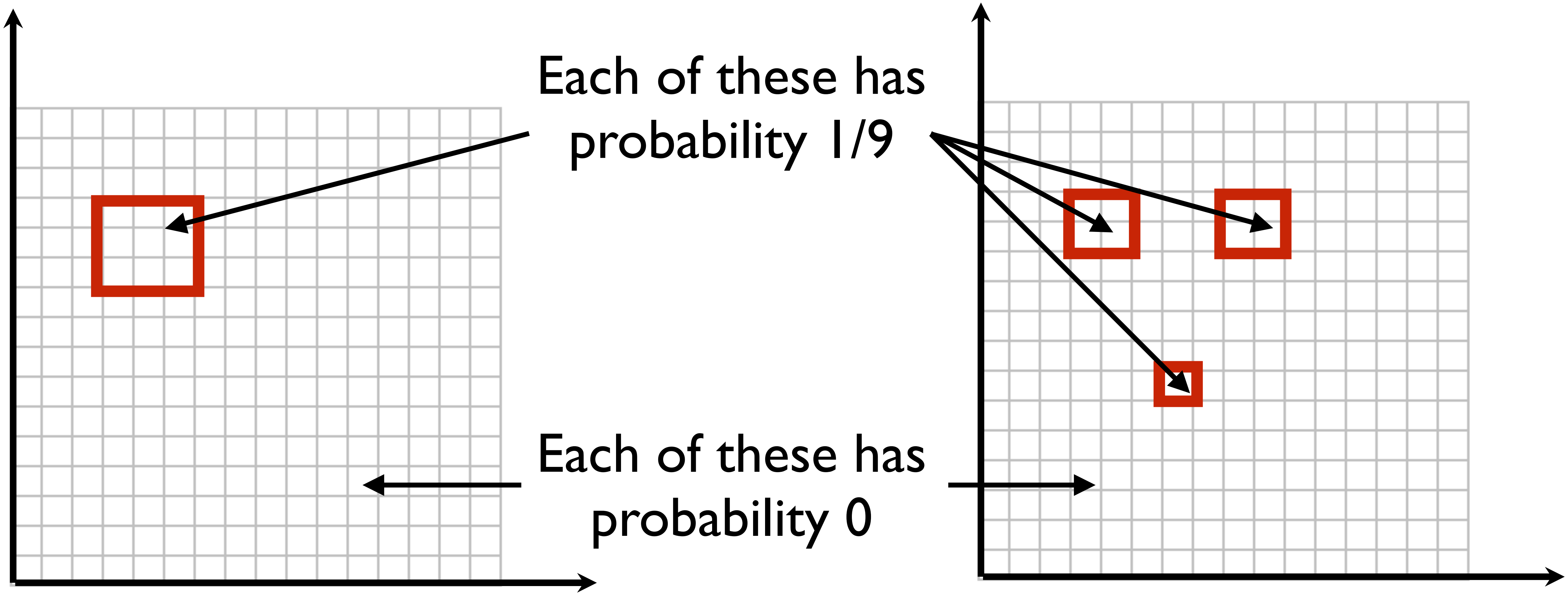
preferred by the prior



Fitting the data: the Bayesian view

$$P(x|h) = \begin{cases} \frac{1}{N_s} & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases}$$

The **likelihood** function assigns probability to data

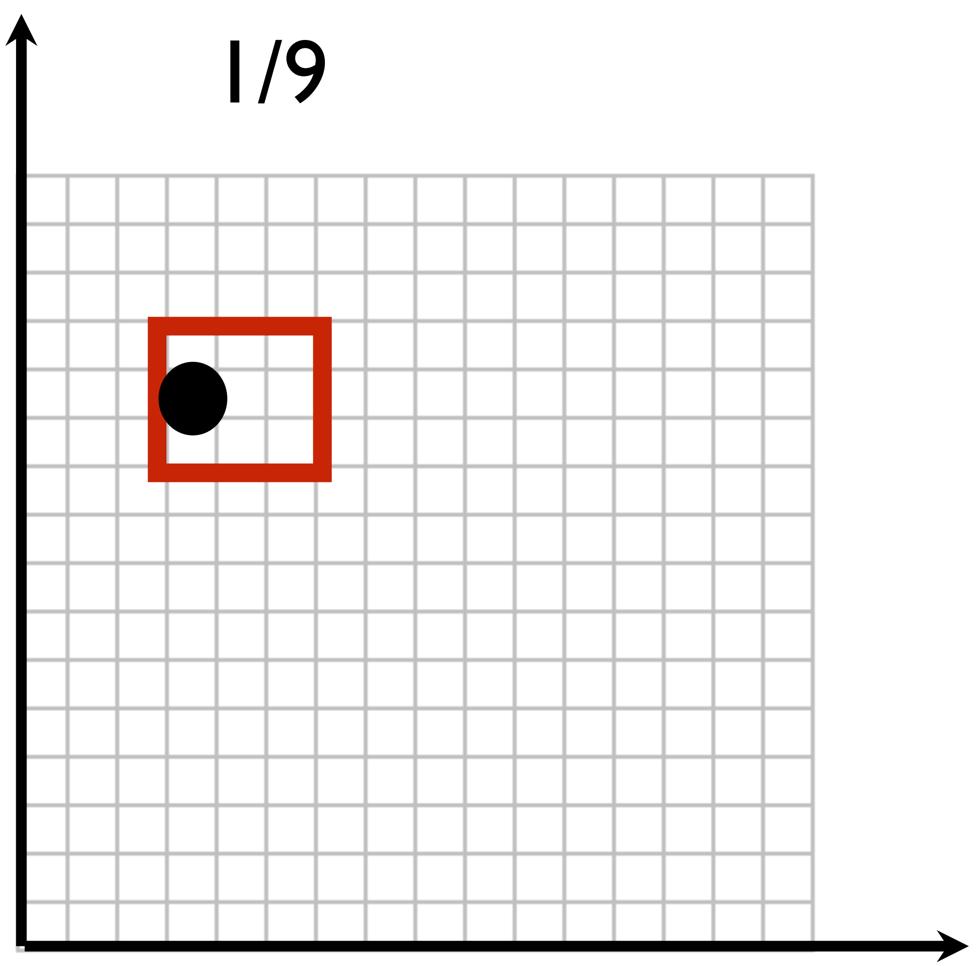


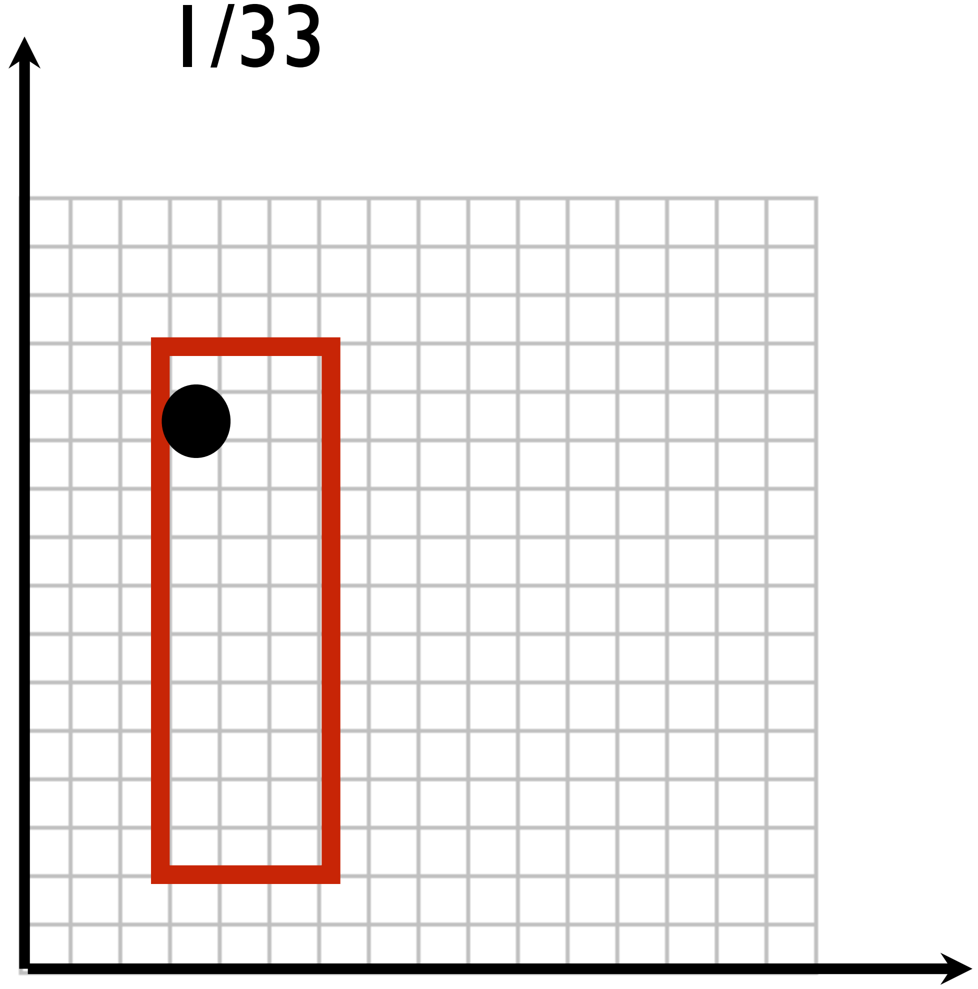
Fitting the data: the Bayesian view

$$P(x|h) = \begin{cases} \frac{1}{N_s} & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases}$$

The likelihood function assigns probability to data

Better fit!

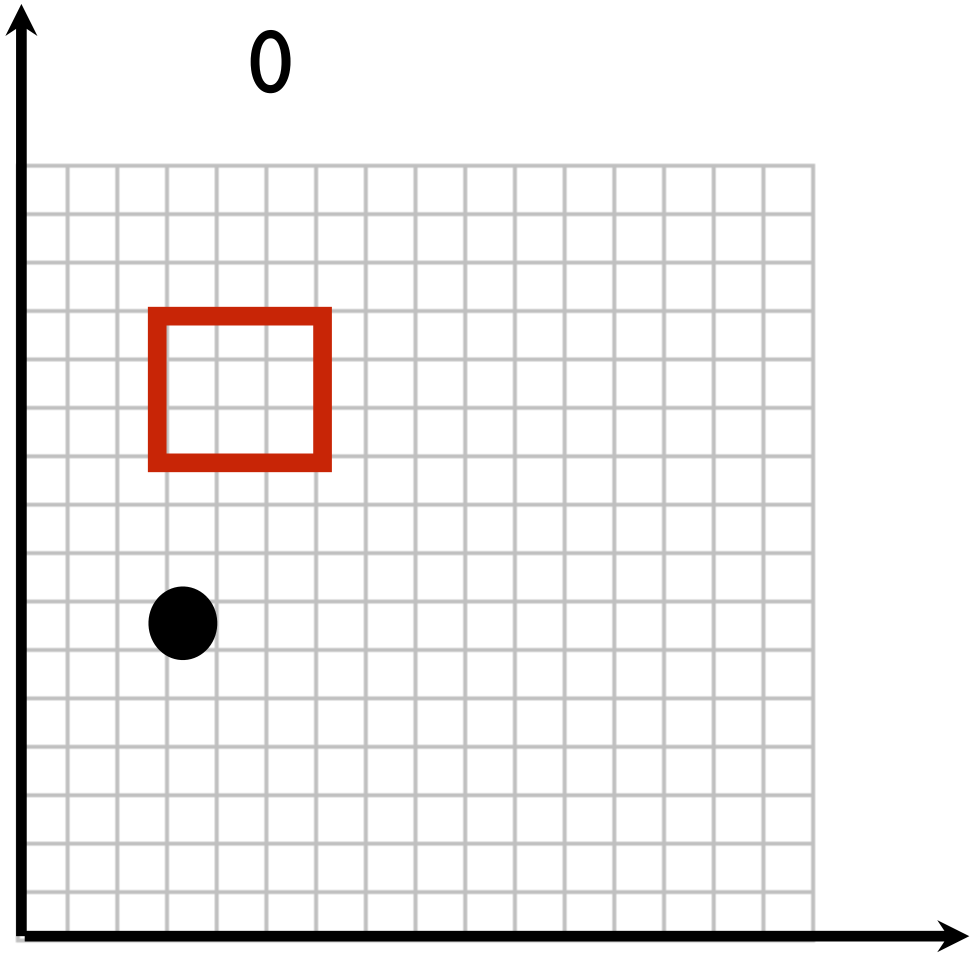


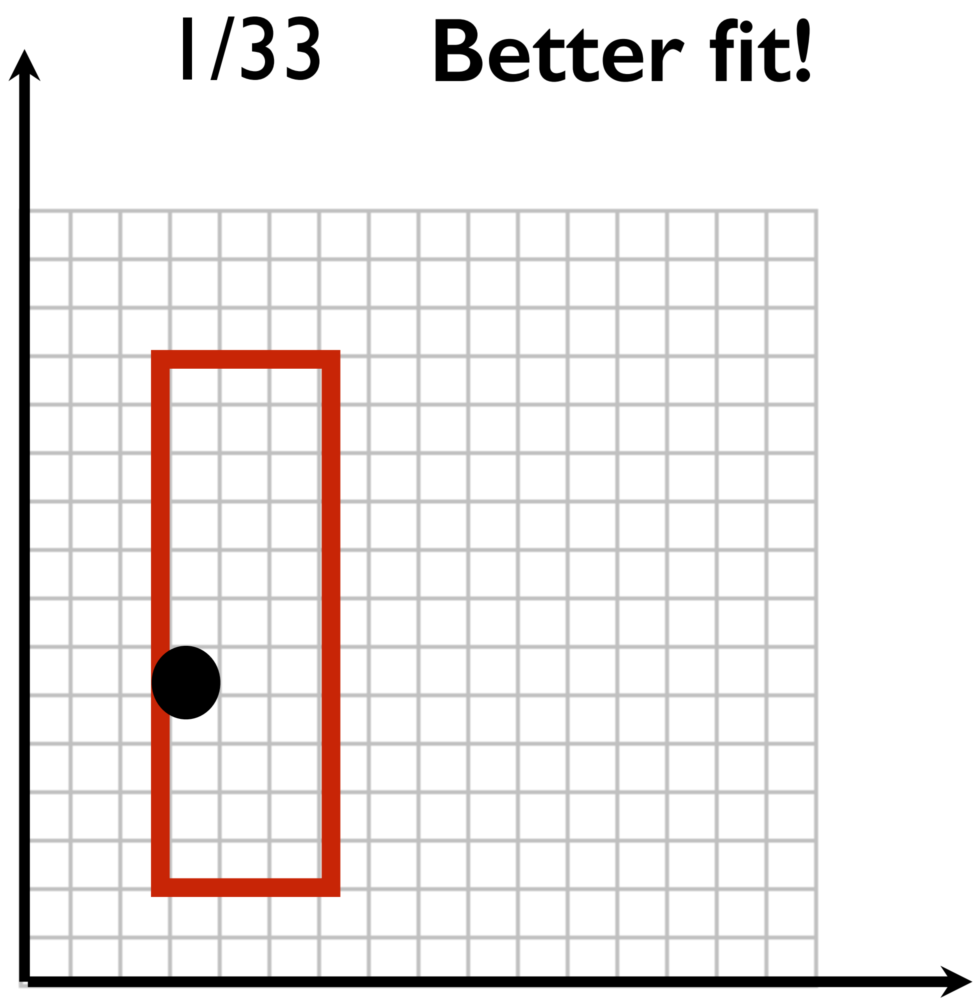


Fitting the data: the Bayesian view

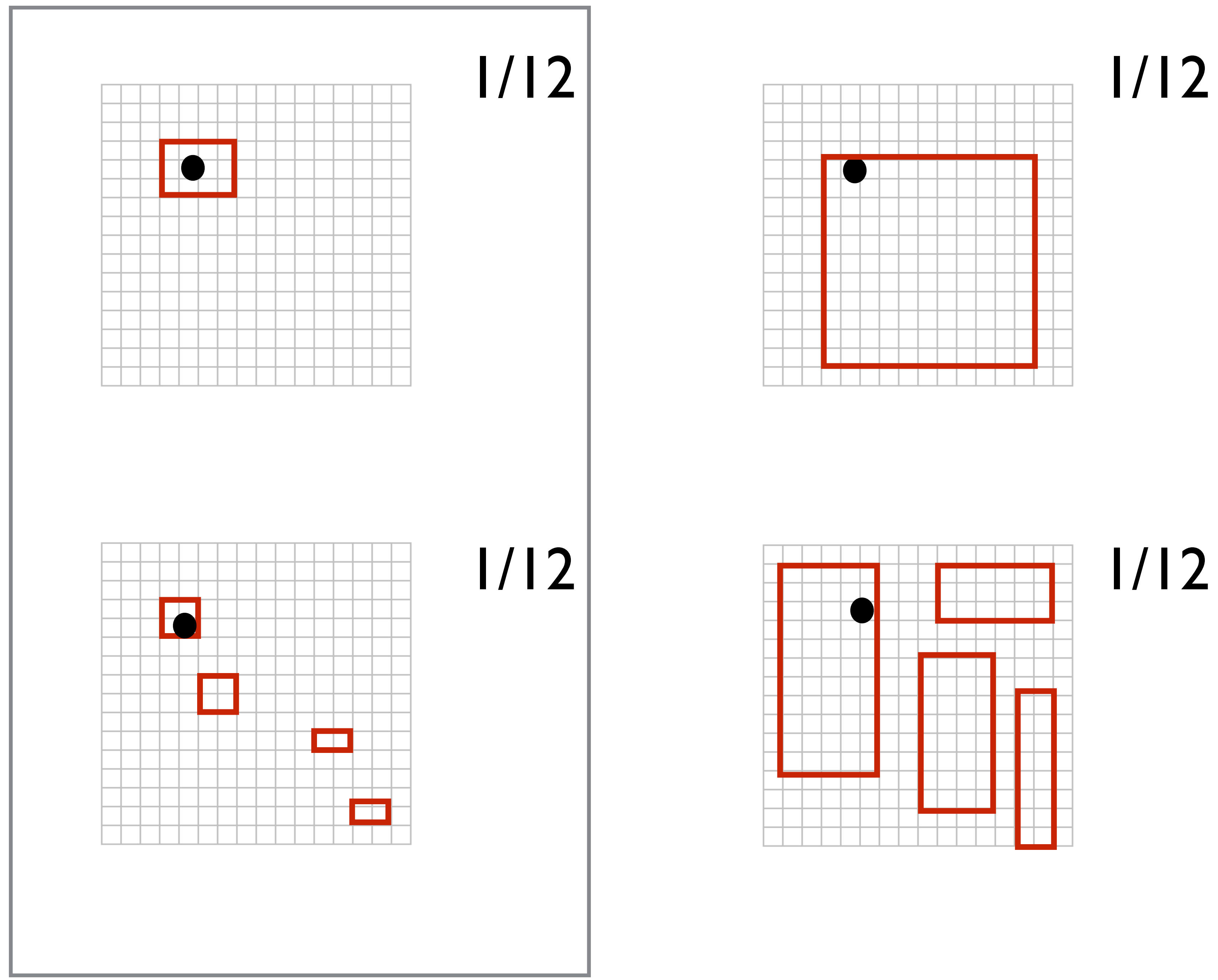
$$P(x|h) = \begin{cases} \frac{1}{N_s} & \text{if } x \in h \\ 0 & \text{otherwise} \end{cases}$$

The likelihood function assigns probability to data





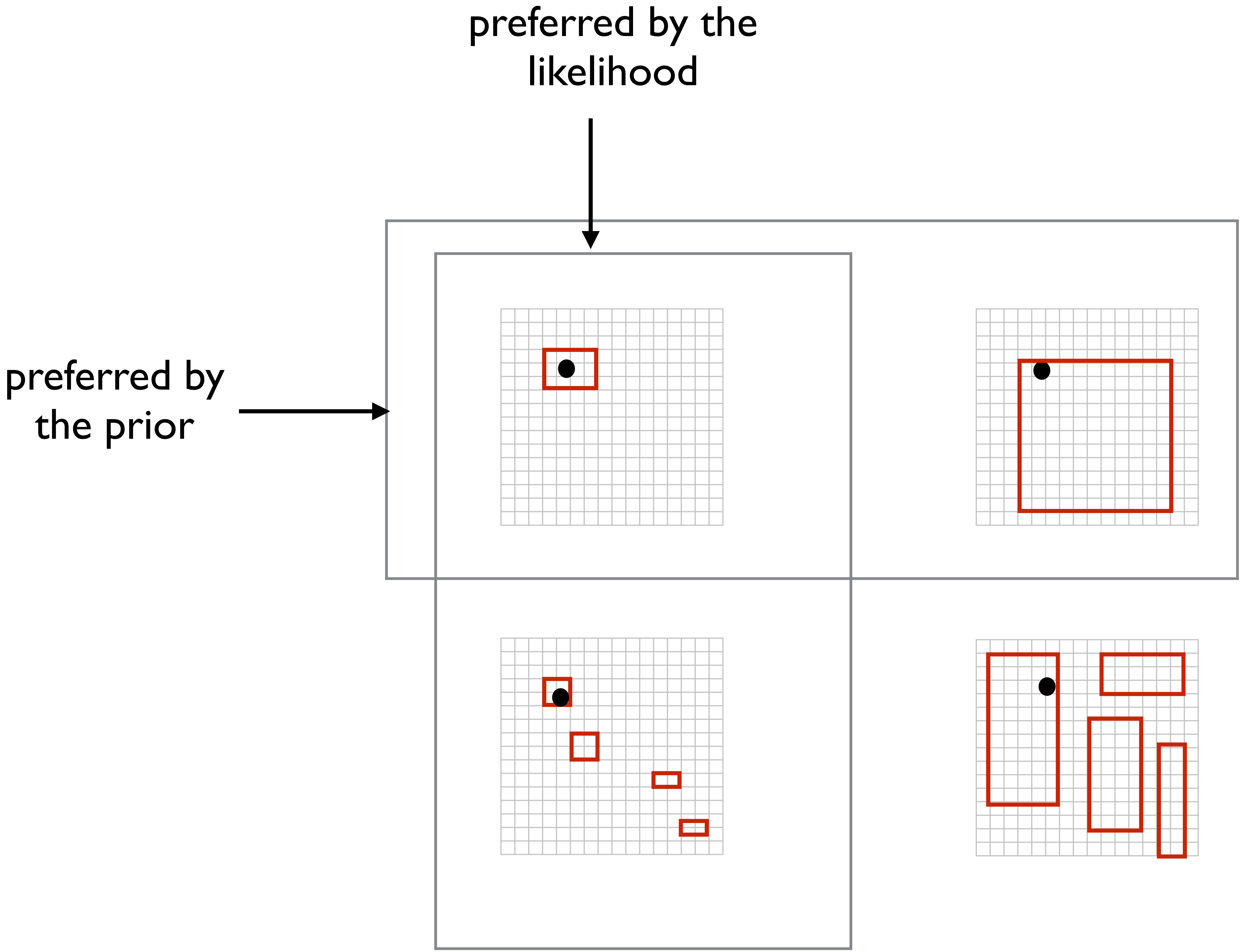
preferred by the likelihood



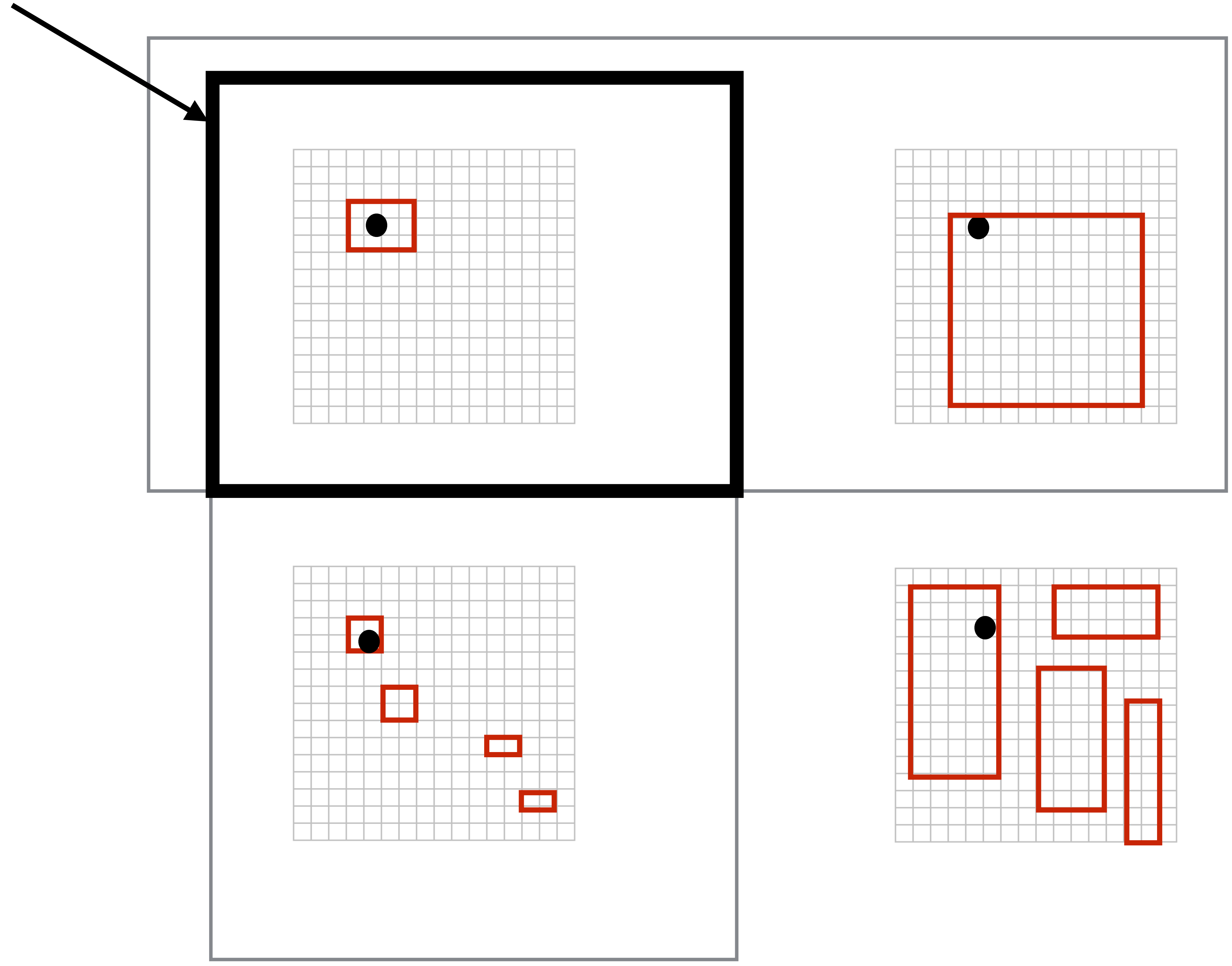
Bayesian Ockham's razor

Likelihood enforces data fit
Prior enforces simplicity
Posterior enforces Ockham's razor

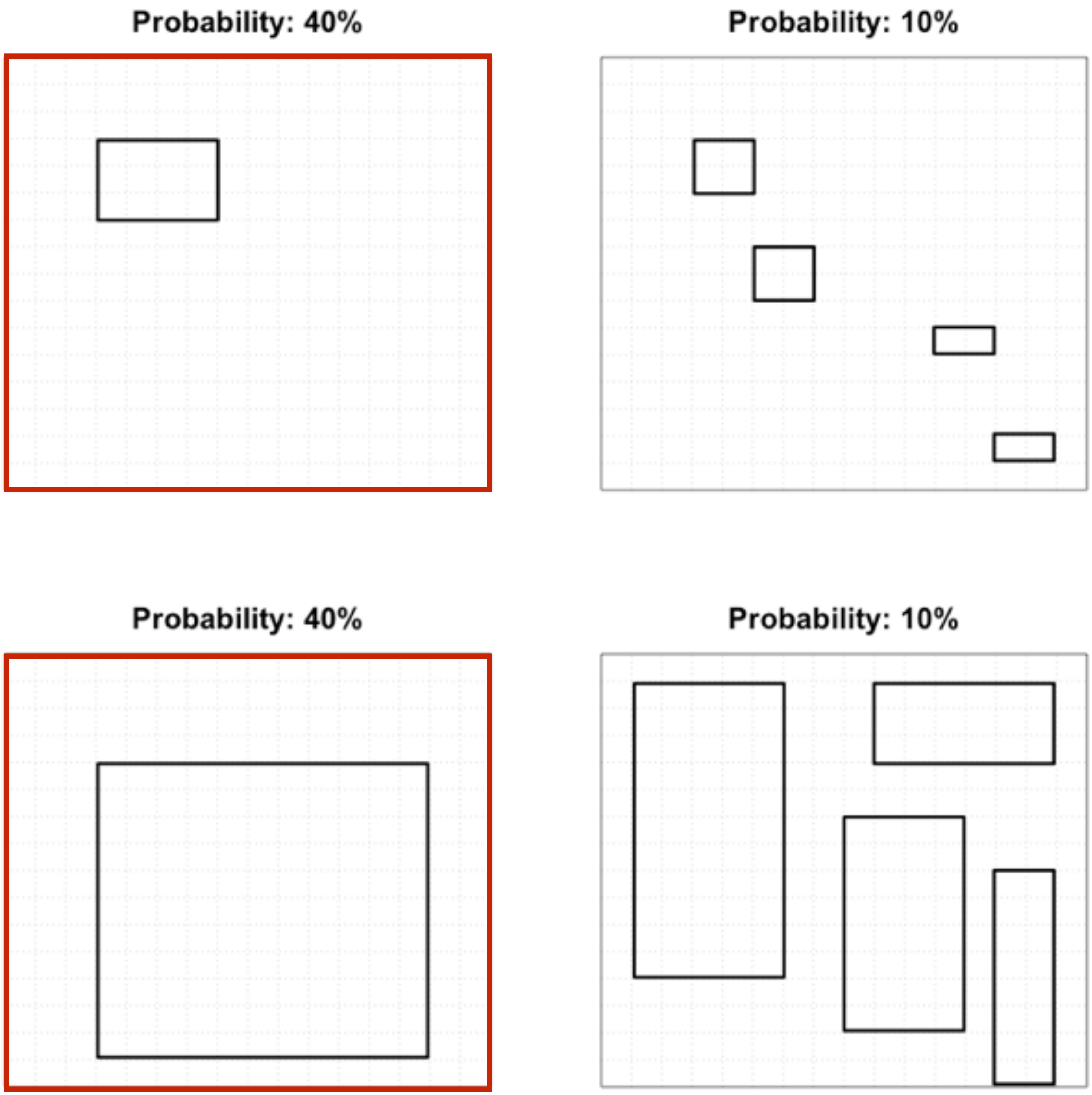
$$
\begin{aligned}
P(h|x) \quad &\propto \quad P(x|h)P(h) \\
&= \quad \frac{1}{N_s} \times \frac{1}{N_e}
\end{aligned}
$$


preferred by the likelihood
preferred by the prior

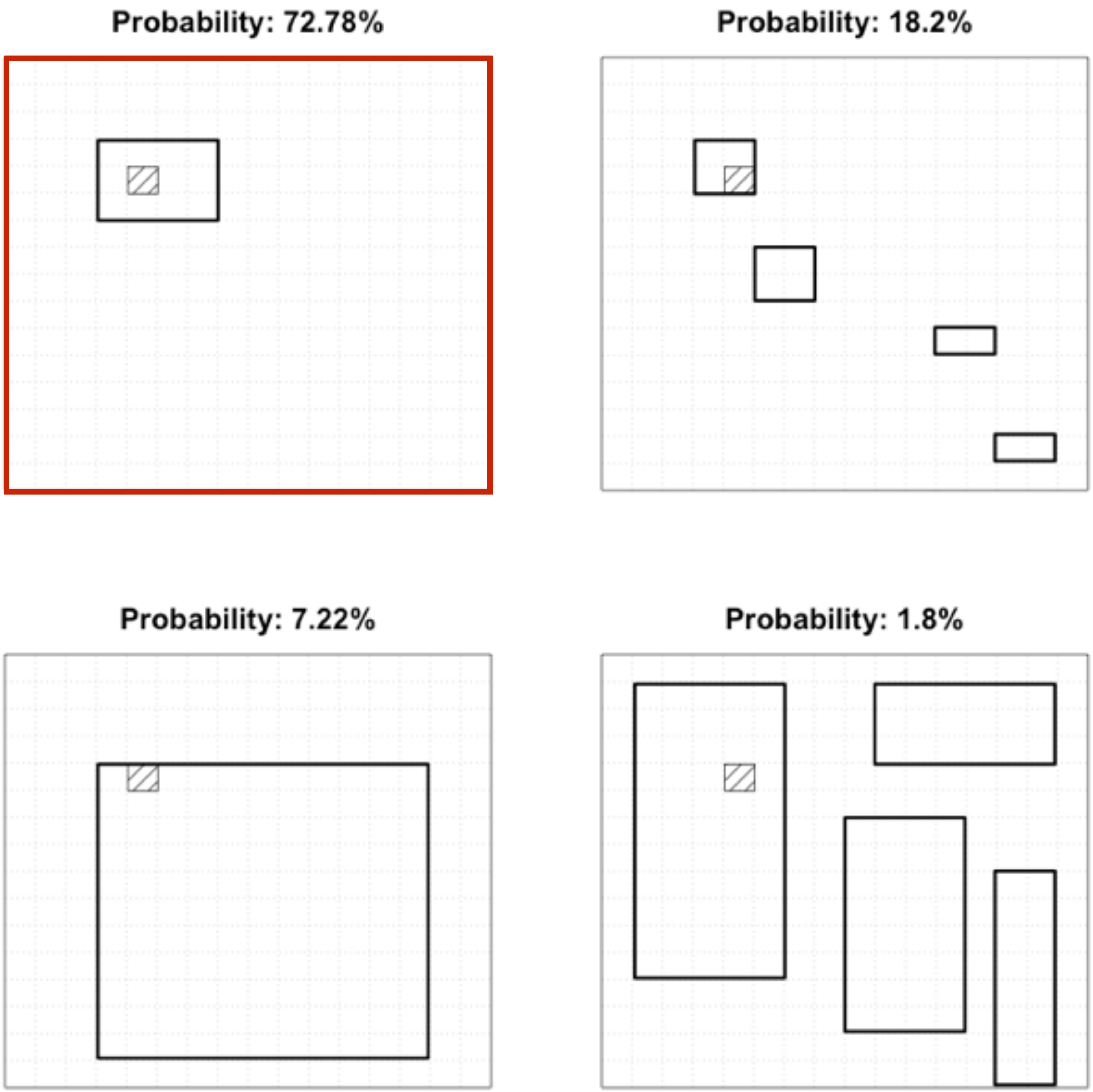
preferred by the posterior



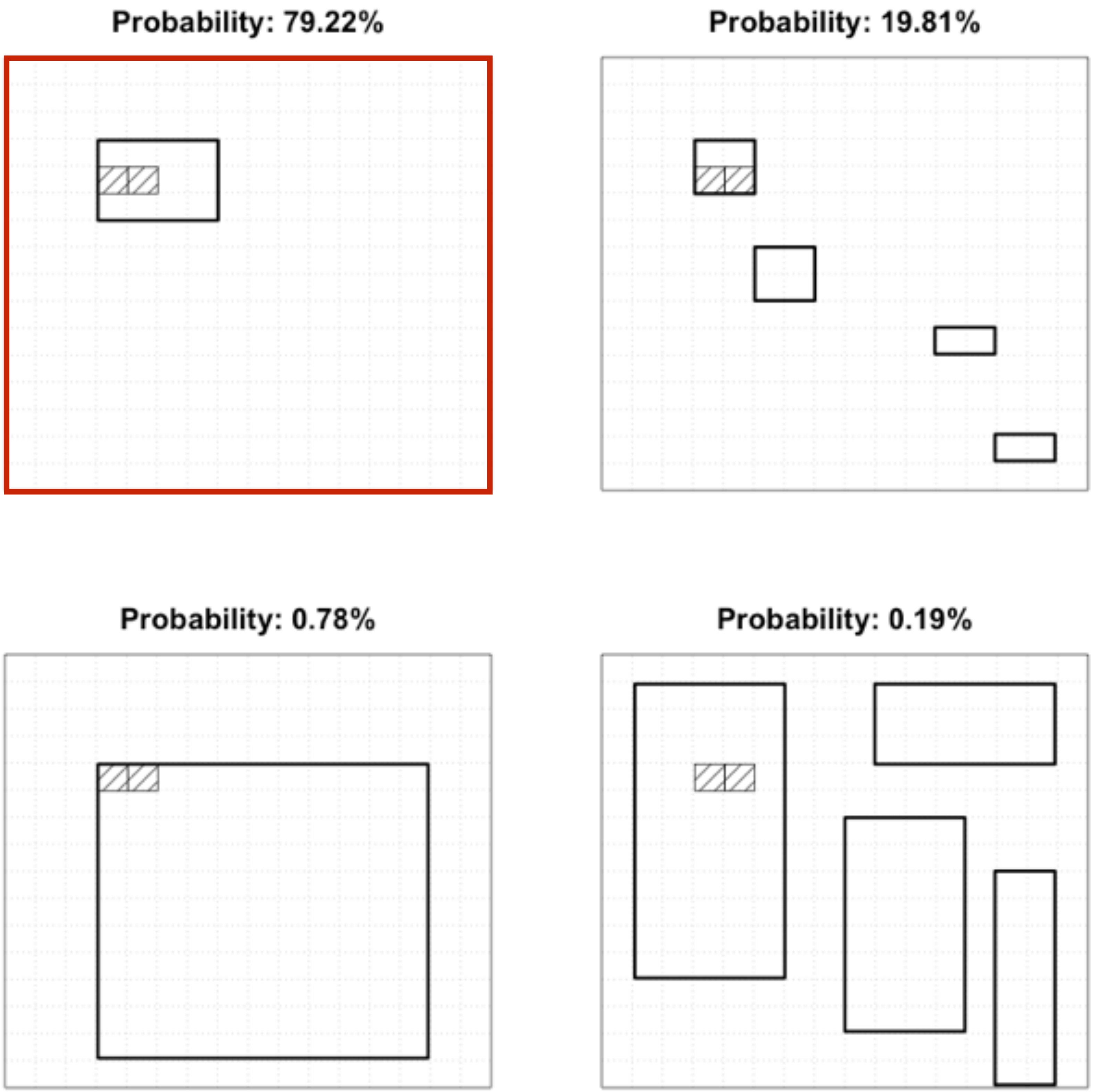
How much is it preferred?
(demo code: battleships1.R)


Probability: 40%
Probability: 10%
Probability: 40%
Probability: 10%

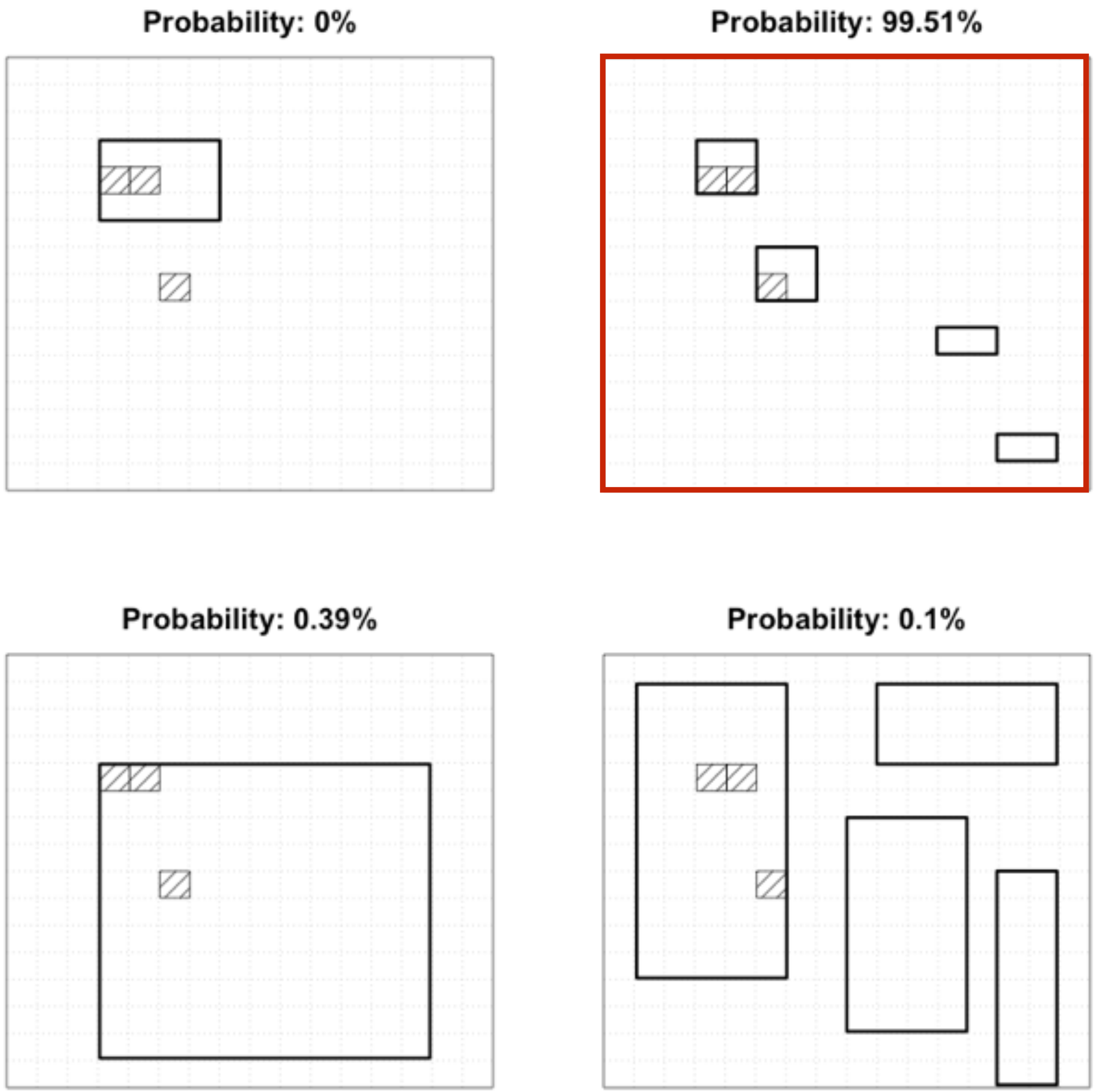
prior



posterior after one observation

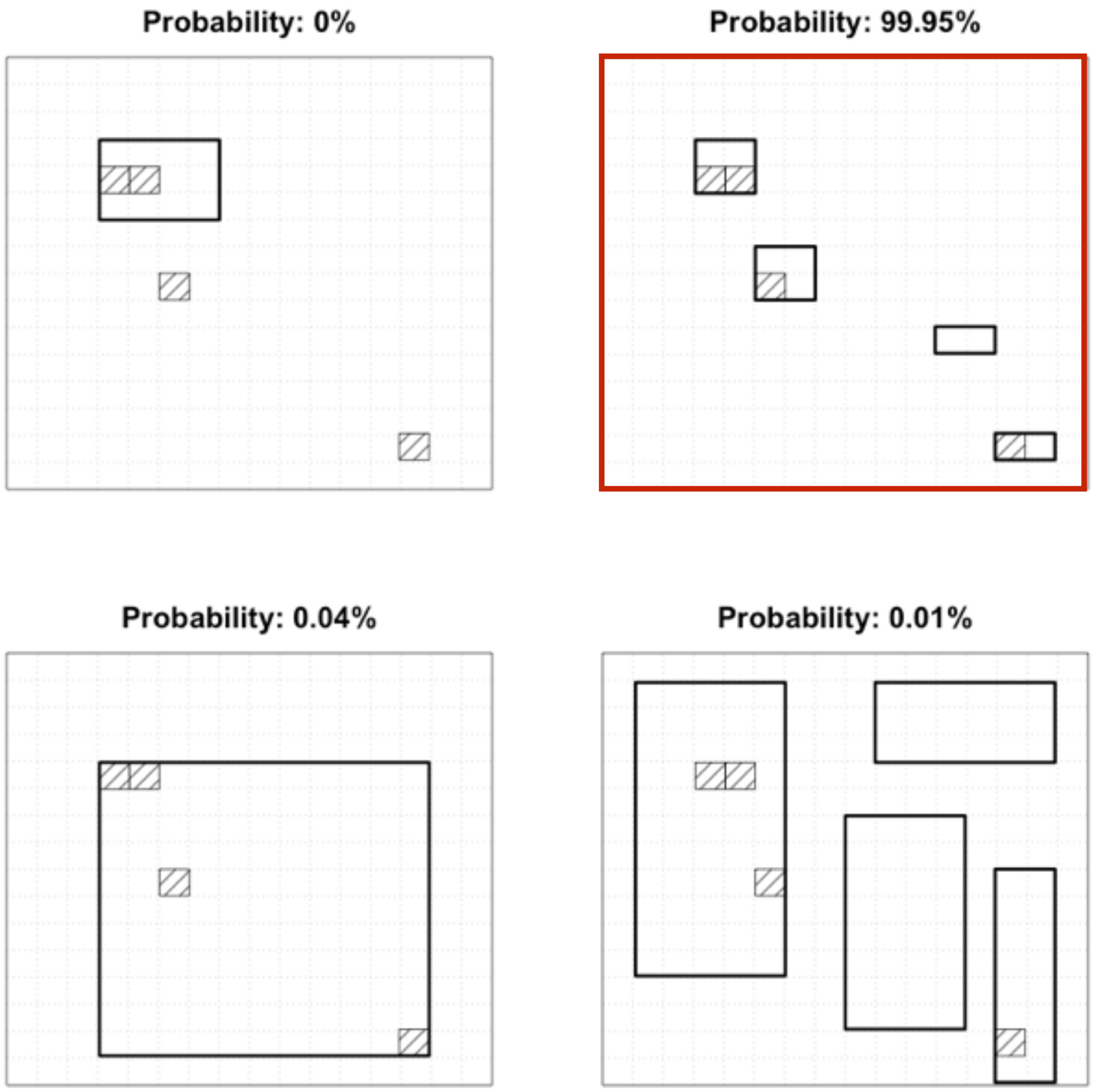


posterior after two observations



posterior after three observations

does 99.5% feel extreme? it should: most people are “conservative” relative to Bayes in this sort of problem



posterior after four observations

All possible 1-ship and 2-ship solutions in a 10x10 grid
(demo code: battleships2.R)

Larger hypothesis space

- In a 10x10 grid, there are:
  - 3025 distinct rectangles
  - 5,009,400 pairs of non-overlapping rectangles

- Simplicity prior: set P(h) so that
  - Total prior probability of 1 rectangle is 67%
  - Total prior probability of 2 rectangles is 33%

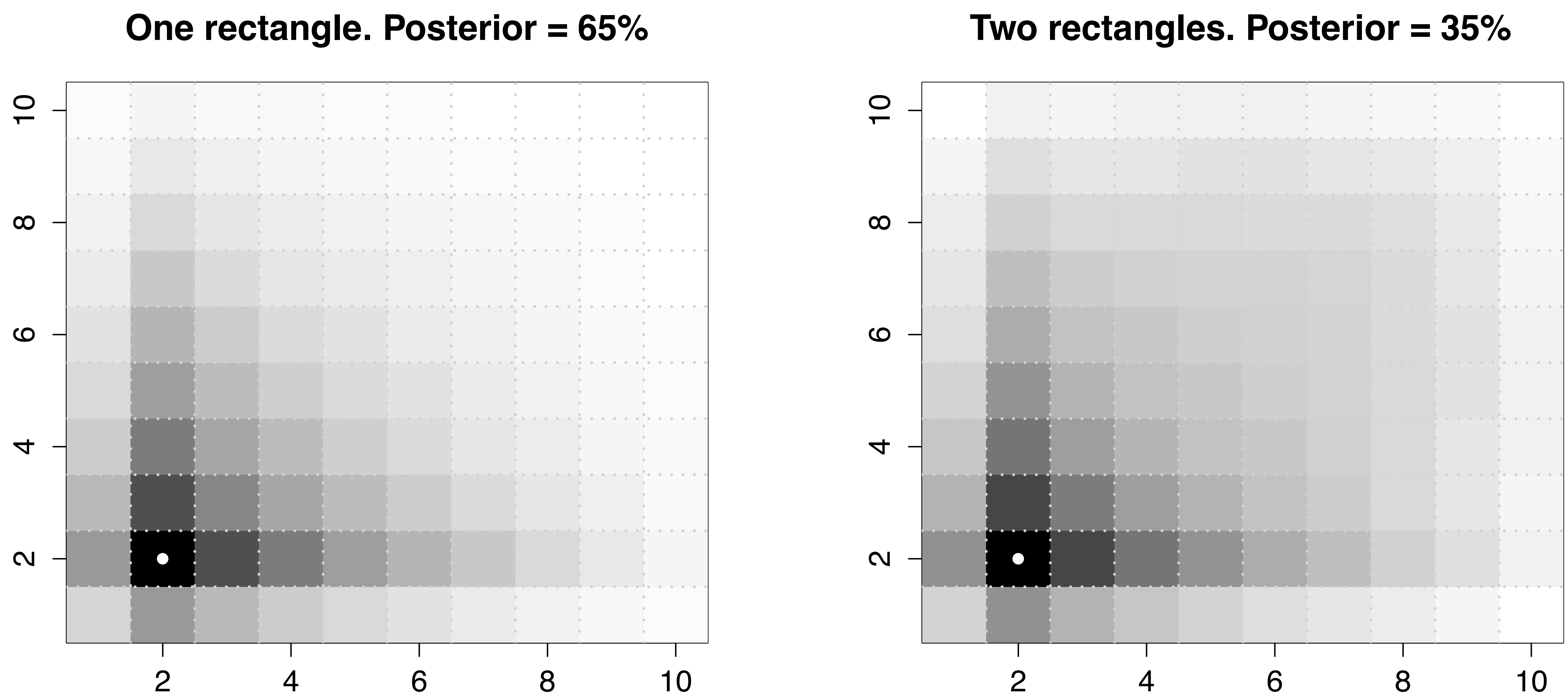
$$P(h) = \frac{1}{3025} \times \frac{2}{3}$$

if h contains one rectangle

$$P(h) = \frac{1}{5009400} \times \frac{1}{3}$$

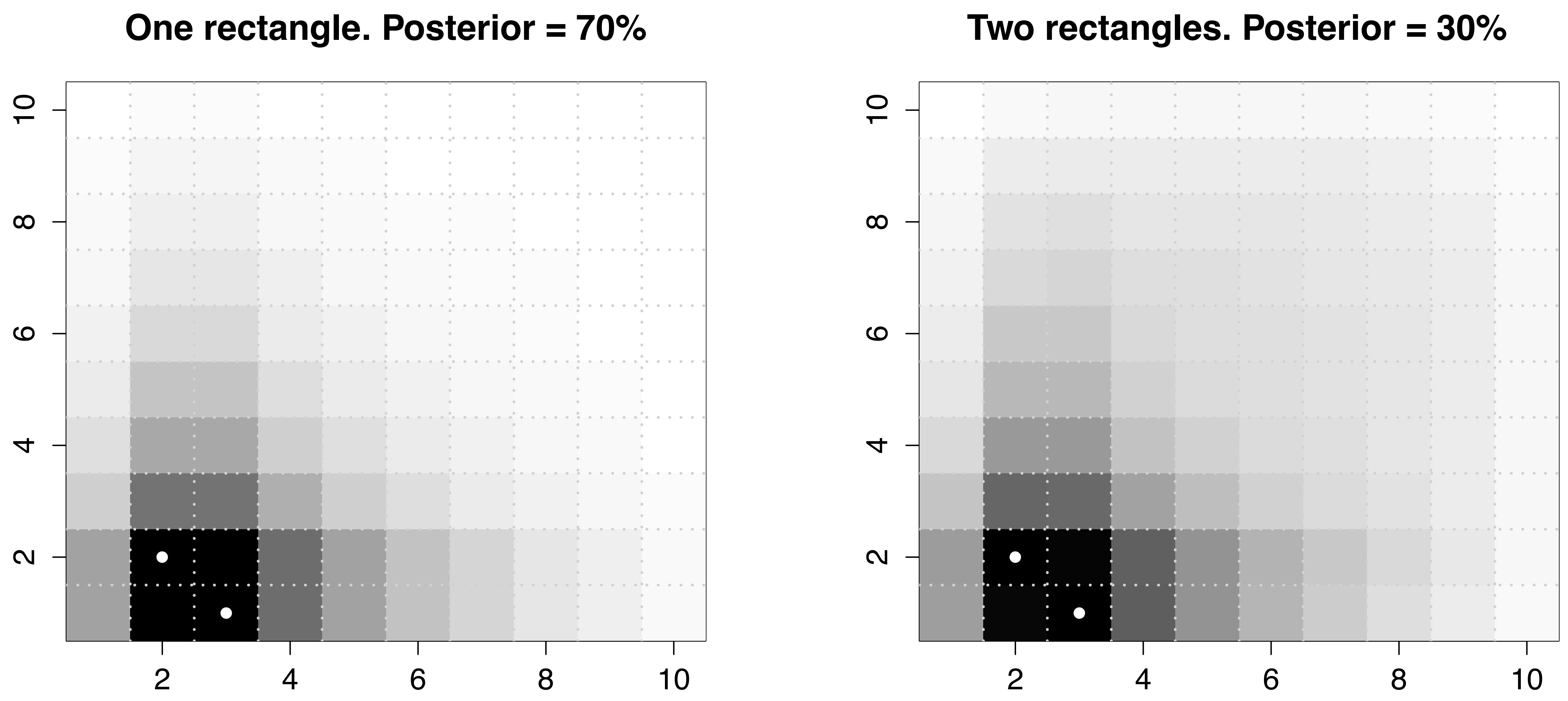
if h contains two rectangles

After one observation

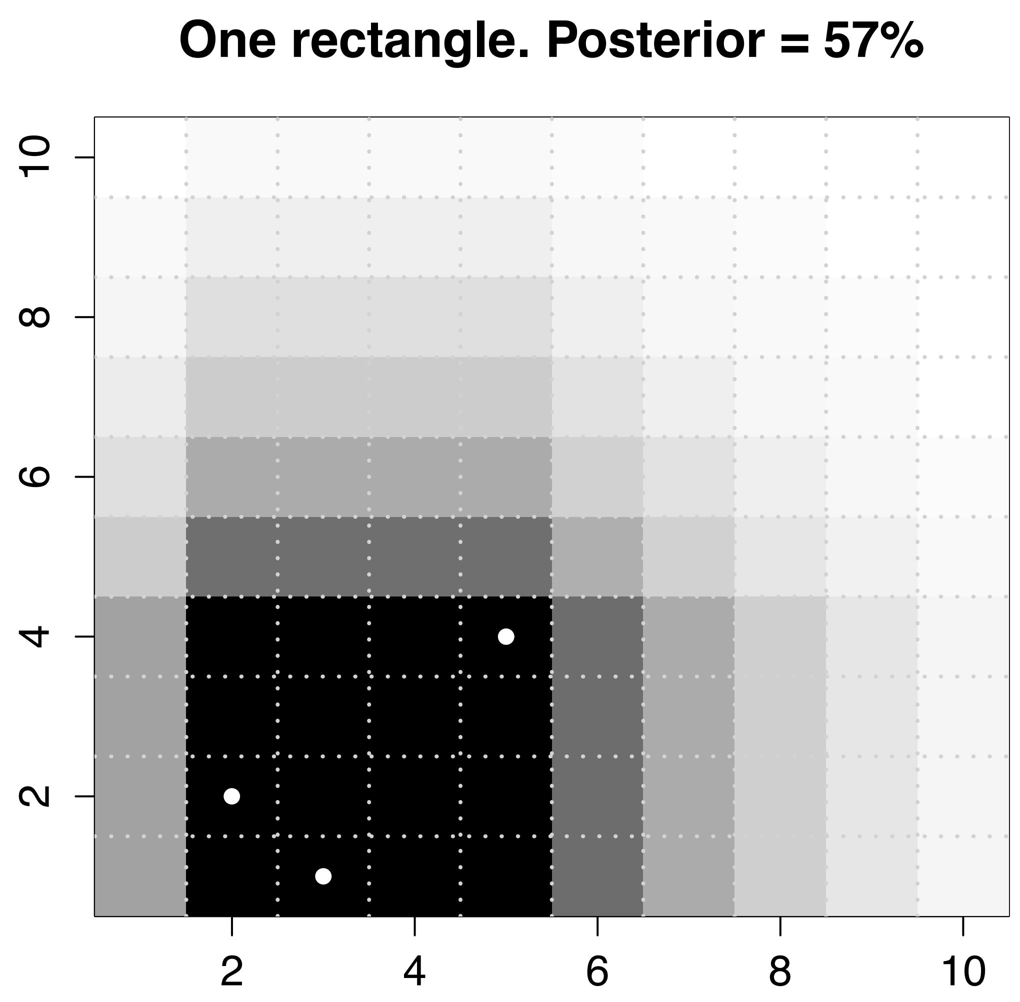


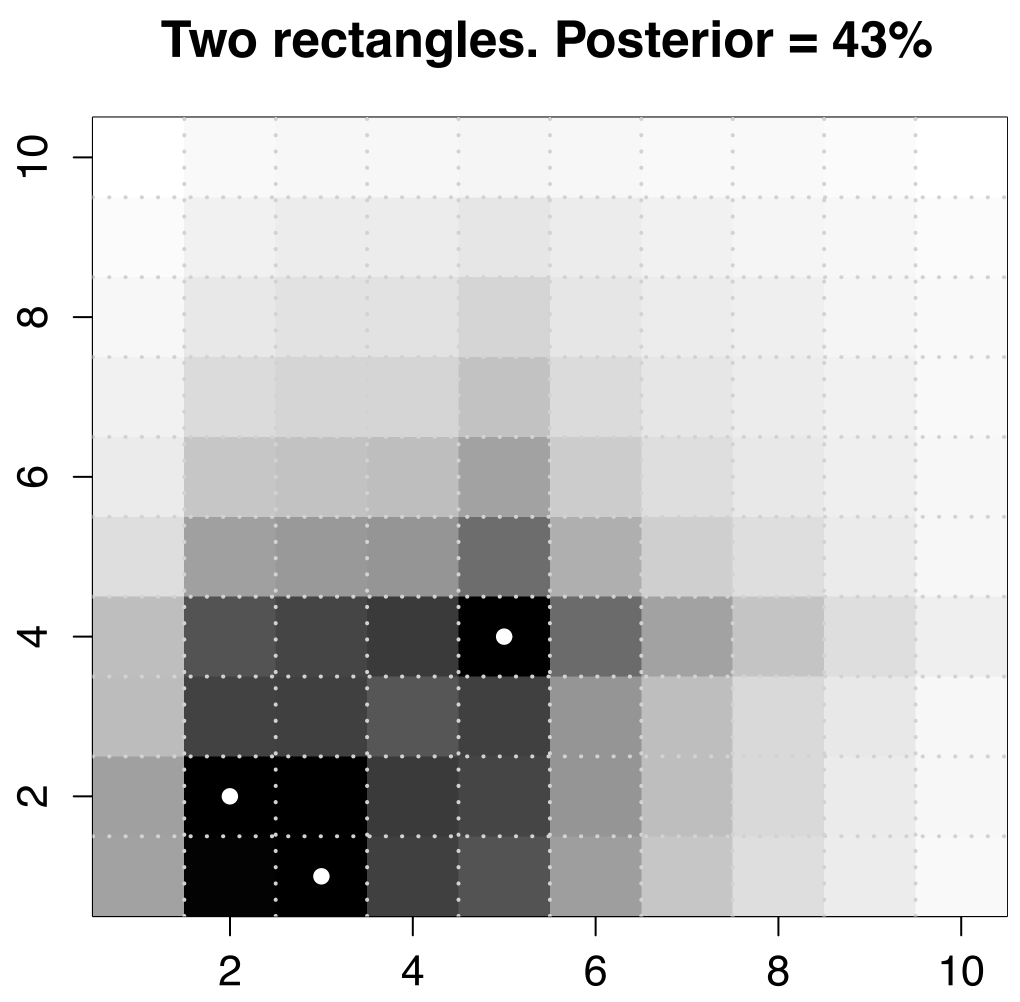
One observation tells you a lot about possible locations (dark squares), but the posterior probability of 1 vs 2 rectangles hasn't moved much from the priors

After two observations



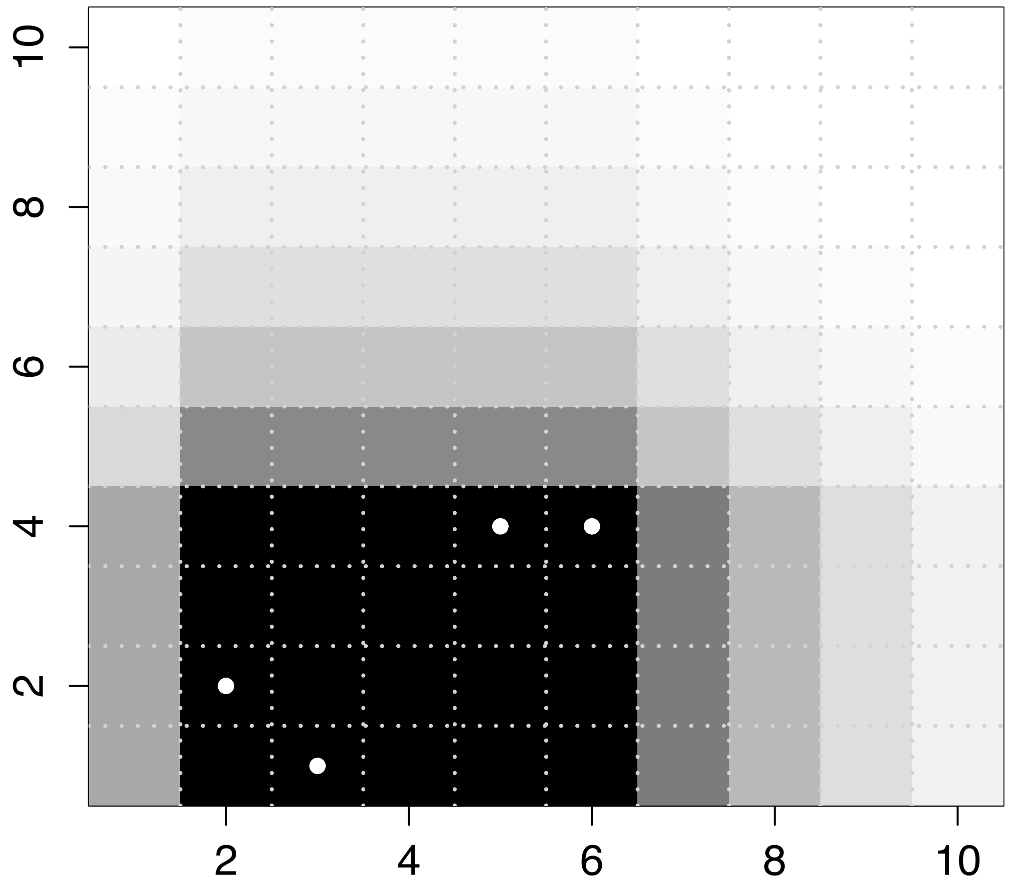
After three observations



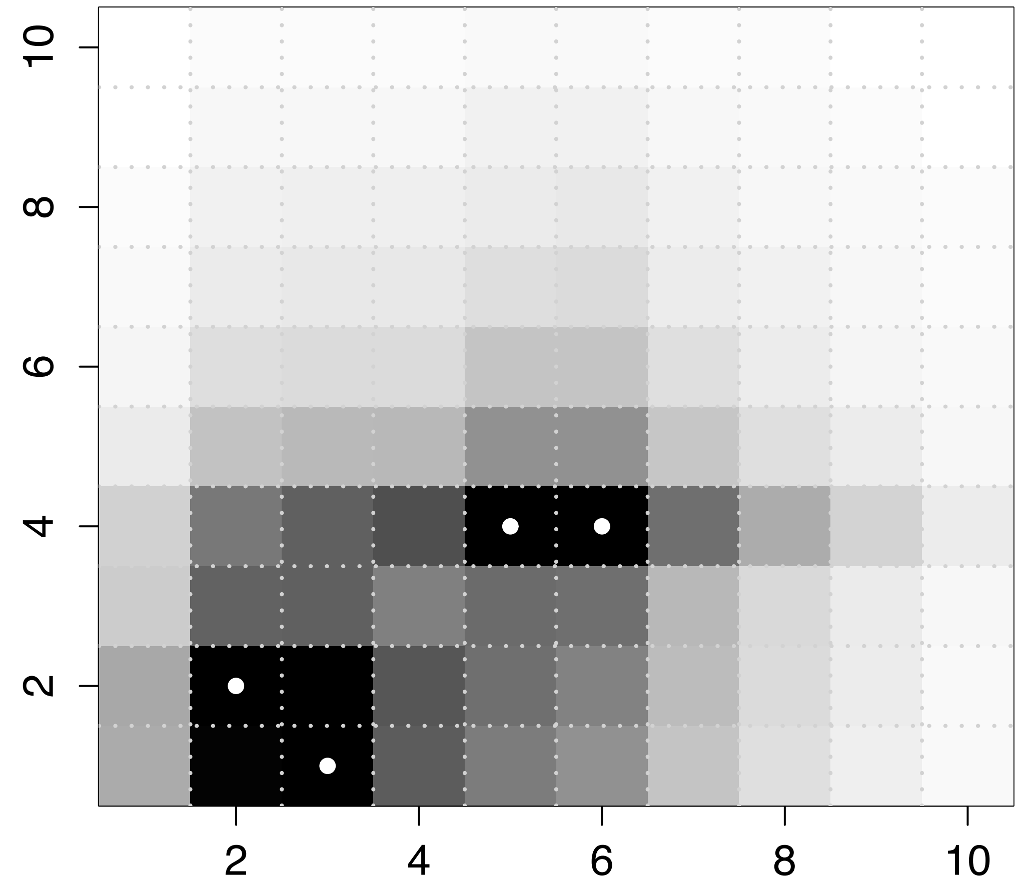


After four observations

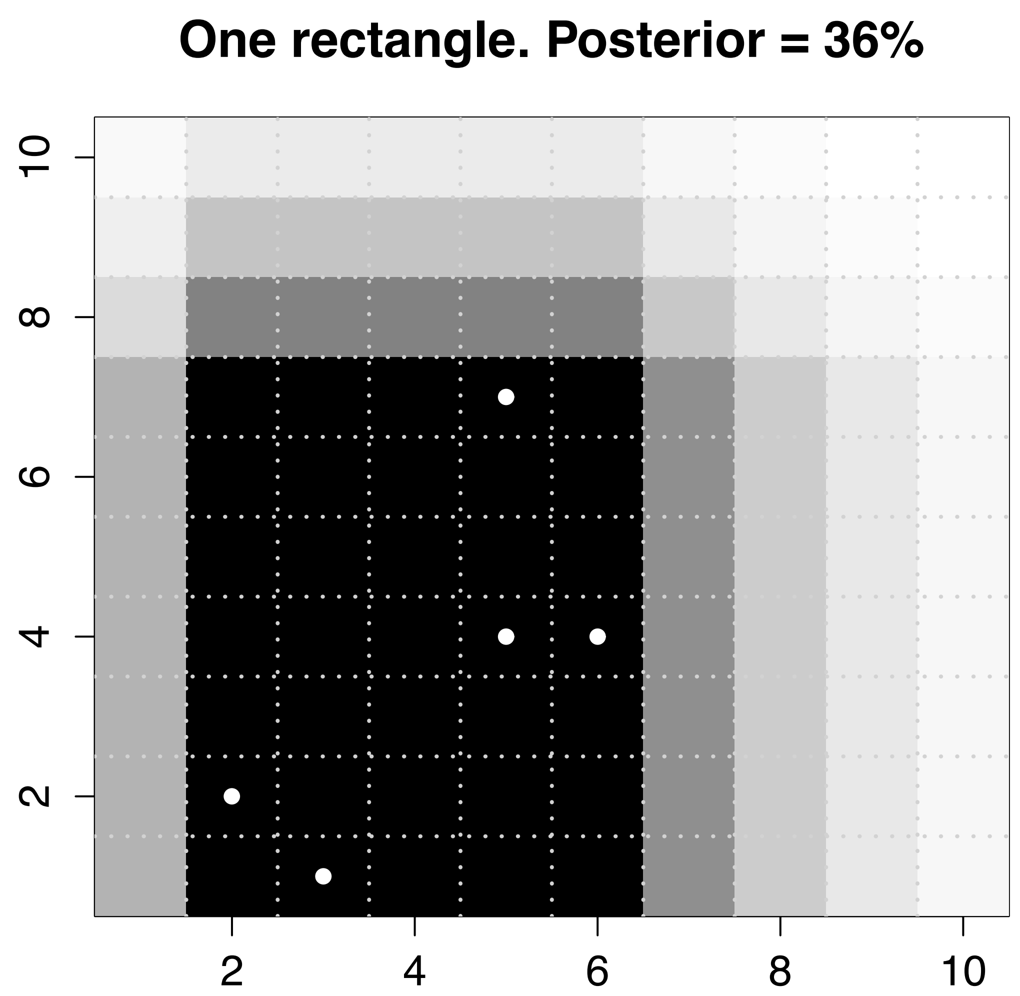
One rectangle. Posterior = 49%

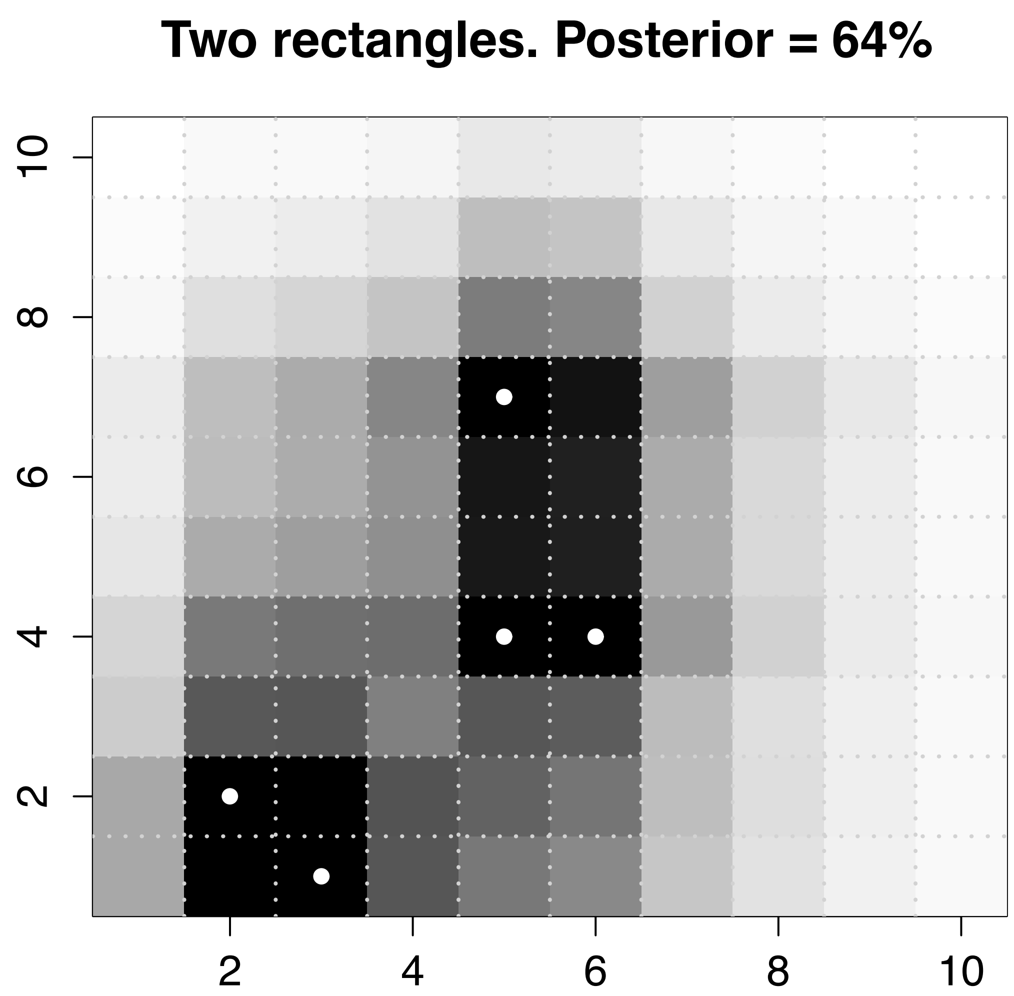


Two rectangles. Posterior = 51%

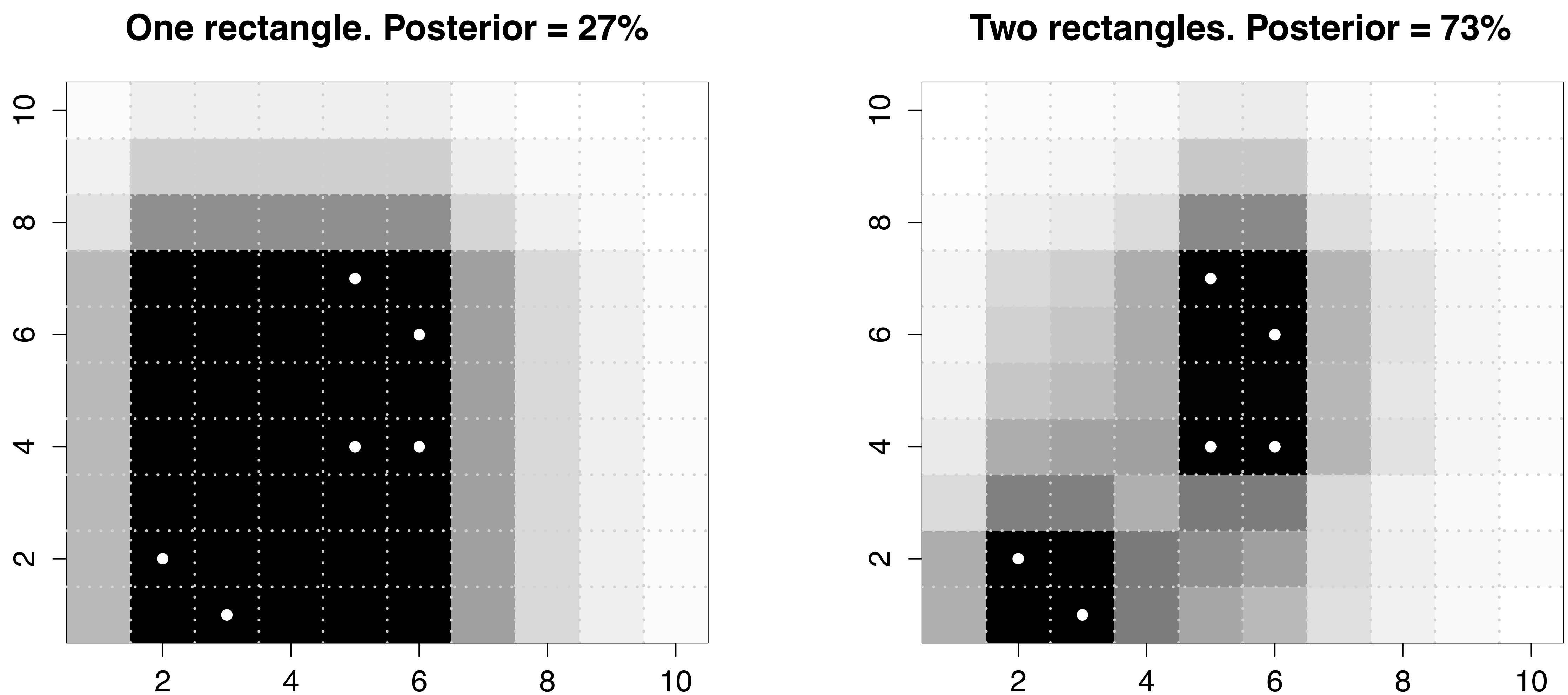


After five observations



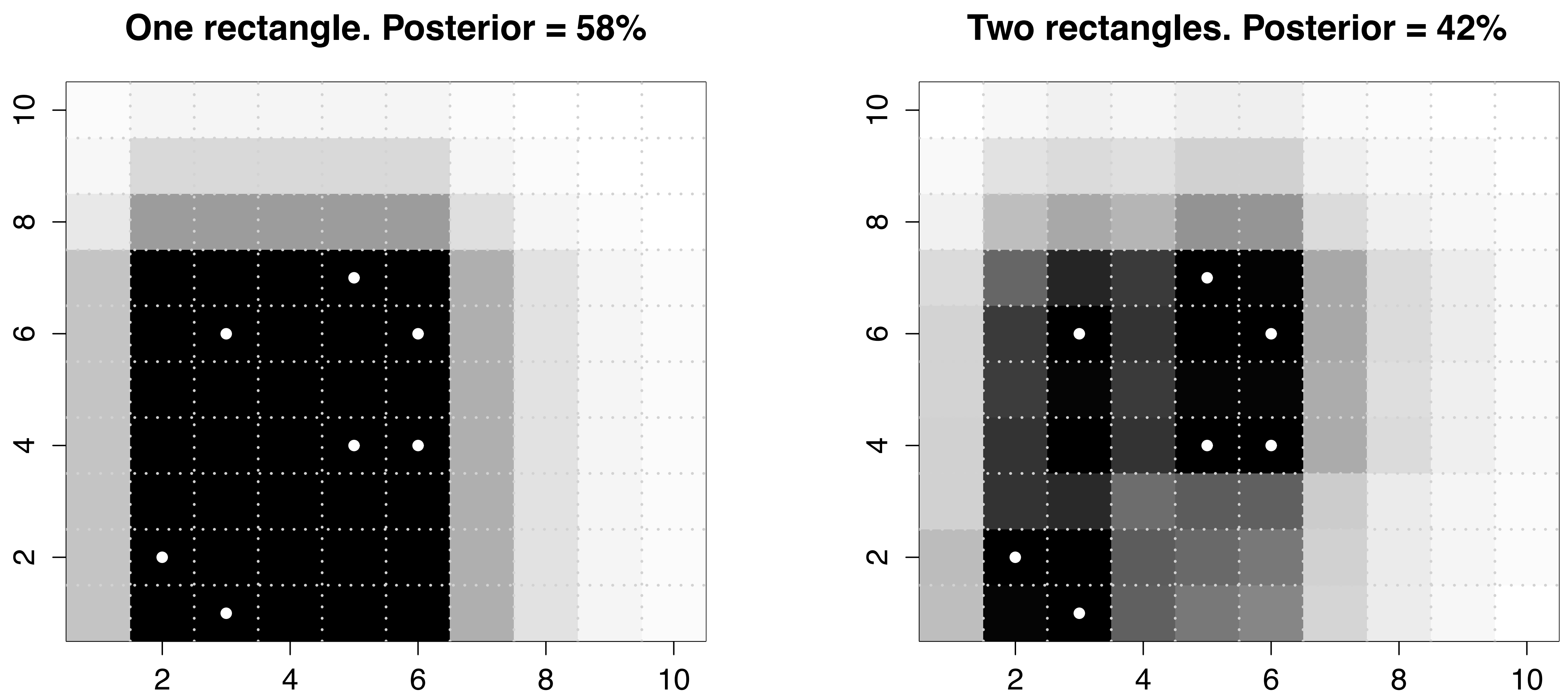


After six observations



At this point the evidence is moderately convincing that there are probably two rectangles here

After seven observations



But it doesn't take much to shift beliefs a long way!

Simplicity from an algorithmic complexity theory perspective

Simplicity = compressability

- Minimum description length principle
  - Simple things are short things
  - Specifically, the more you can compress something (using some "sensible" algorithm), the simpler it is

complex	simple
1001010111011	1111111111111
1011101011011	1111111111111
1111111111010	1111111111111
1001111010011	0000000000000
1100110110011	0000000000000

The idealised version

- Kolmogorov complexity
  - The complexity $K(s)$ of string s with respect to programming language L is the length in bits of the shortest program that prints s and then halts
  - The language L doesn't actually matter much
  - The tricky part is that $K(s)$ is uncomputable

- Solomonoff's universal prior
  - Each hypothesis is encoded as a string h
  - Optimal version of Ockham's razor uses the prior:

$$P(h) \propto 2^{-K(h)}$$

Various practical suggestions

- Use a small set of Turing machines, instead of considering all possible programs written for a universal Turing machine (Dowe, Wallace)
- Use statistical considerations to figure out what prior minimises your worst-case loss (Rissanen)
- Use a real compression algorithm to do the work (e.g. Lempel-Ziv-Welch)
- Use something that intuitively seems to capture the idea of simplicity (most of us!)