Natural Language Processing

Lecture 07

Dirk Hovy

dirk.hovy@unibocconi.it

@dirk_hovy

Bocconi

Goals for Today

- Understand the probabilistic approach to language
- Learn how to compute *n*-gram probabilities with **Maximum Likelihood Estimation**
- Understand the **Markov assumption**
- Understand the effect of **smoothing**
- Learn how to use probabilistic language models for **inference** and **generation**

Bocconi

Ranking Sentences

LANGUAGE MODEL

P(S)

- I love to models language `0.48`
- I love language models `0.62`
- I love to language model `0.23`

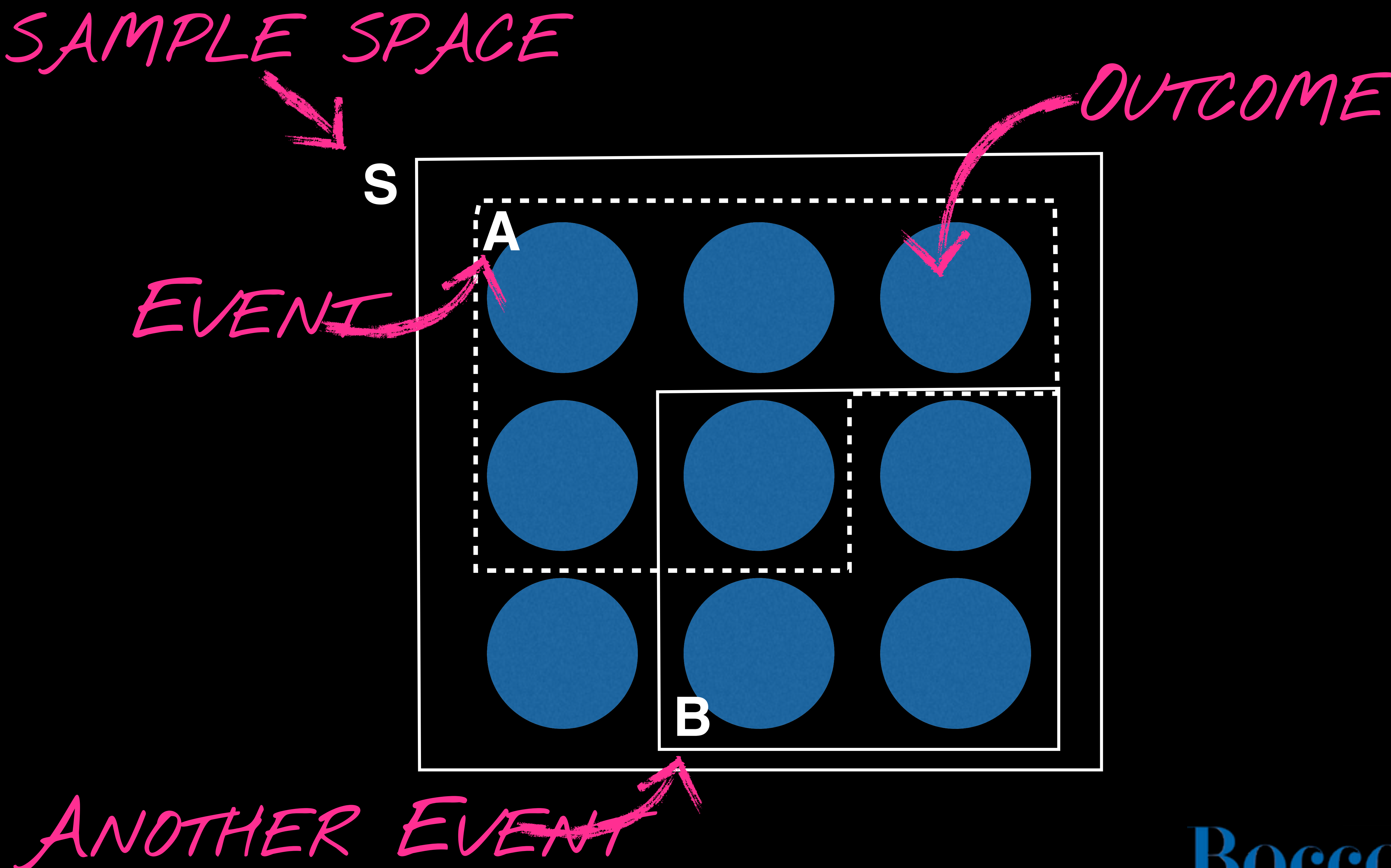
MOST LIKELY TO BE OBSERVED

Language Model in Short

1. Break sentence into *n*-grams
2. Increase their counts
3. Compute probabilities
4. Multiply them together

Recap Probability

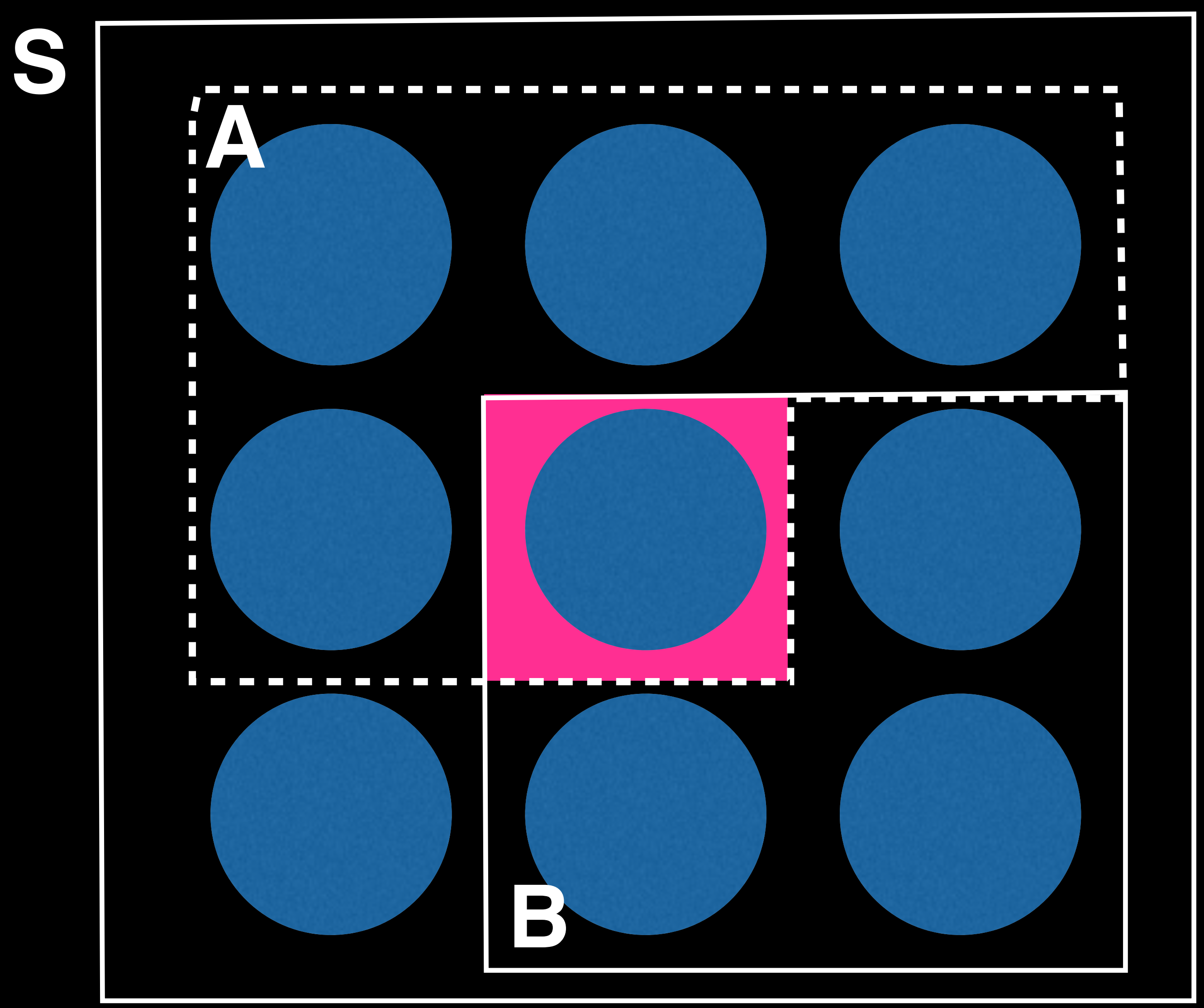
Sample space & Pebbles



Bocconi

Joint Probabilities

INTERSECTION

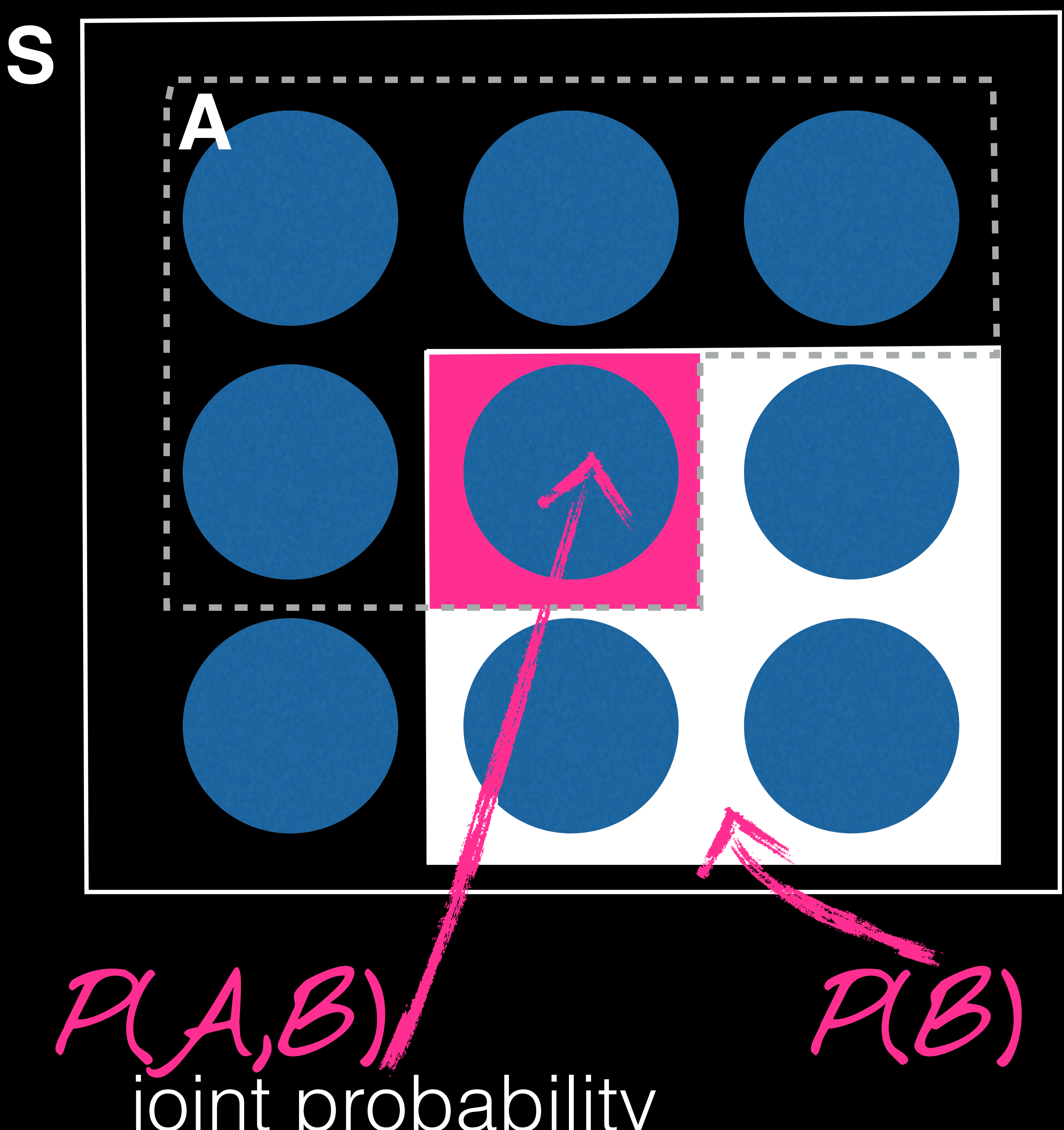


Sets: $A \cap B$

Probability: $P(A,B)$

Meaning: "AND", joint probability

Conditional probability

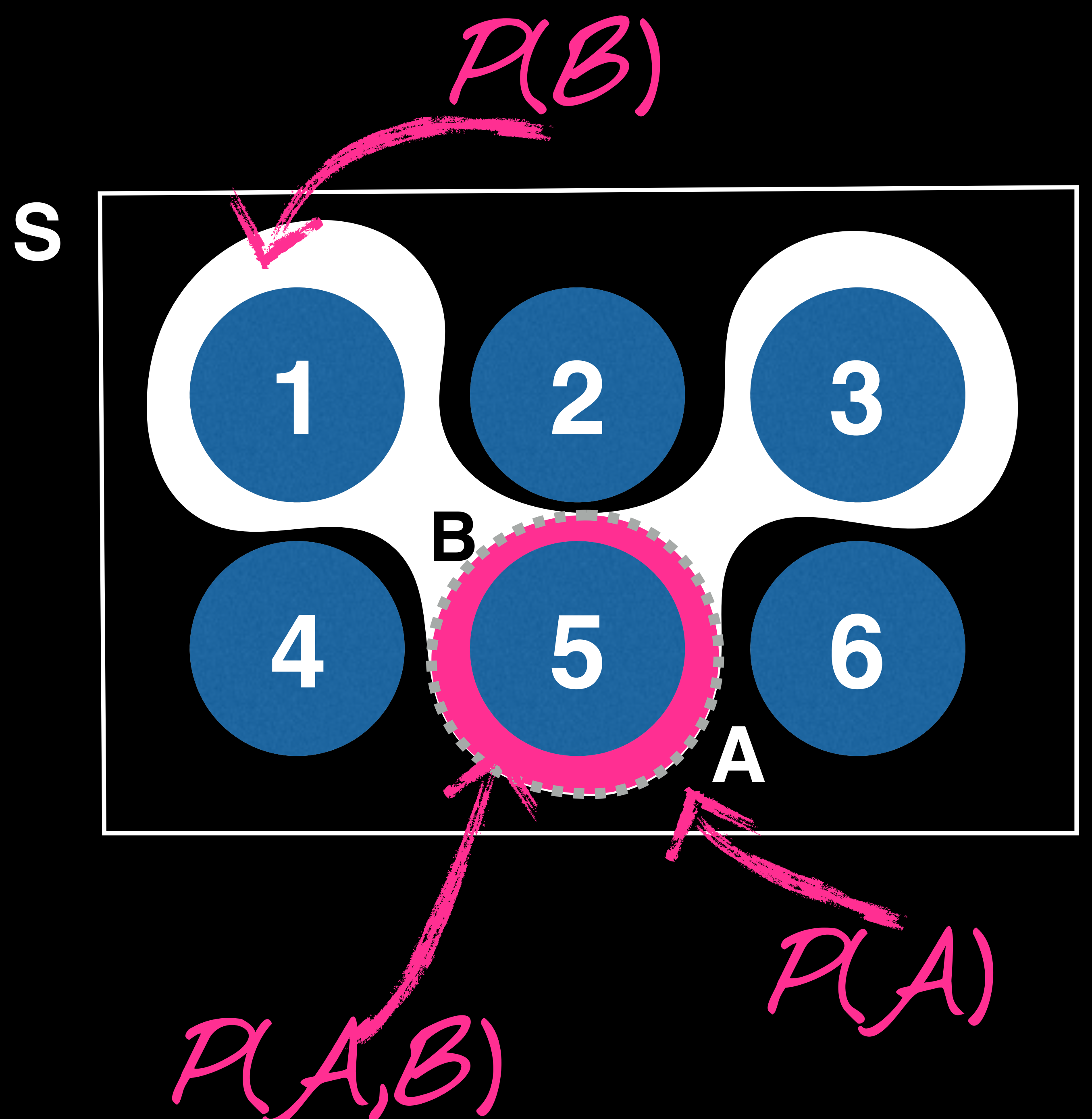


$$P(A|B) = \frac{P(A, B)}{P(B)}$$

$$= \frac{1}{4} = 0.25$$

Bocconi

Conditional probability



A = "get a 5" 1/6

B = "odd number" 3/6

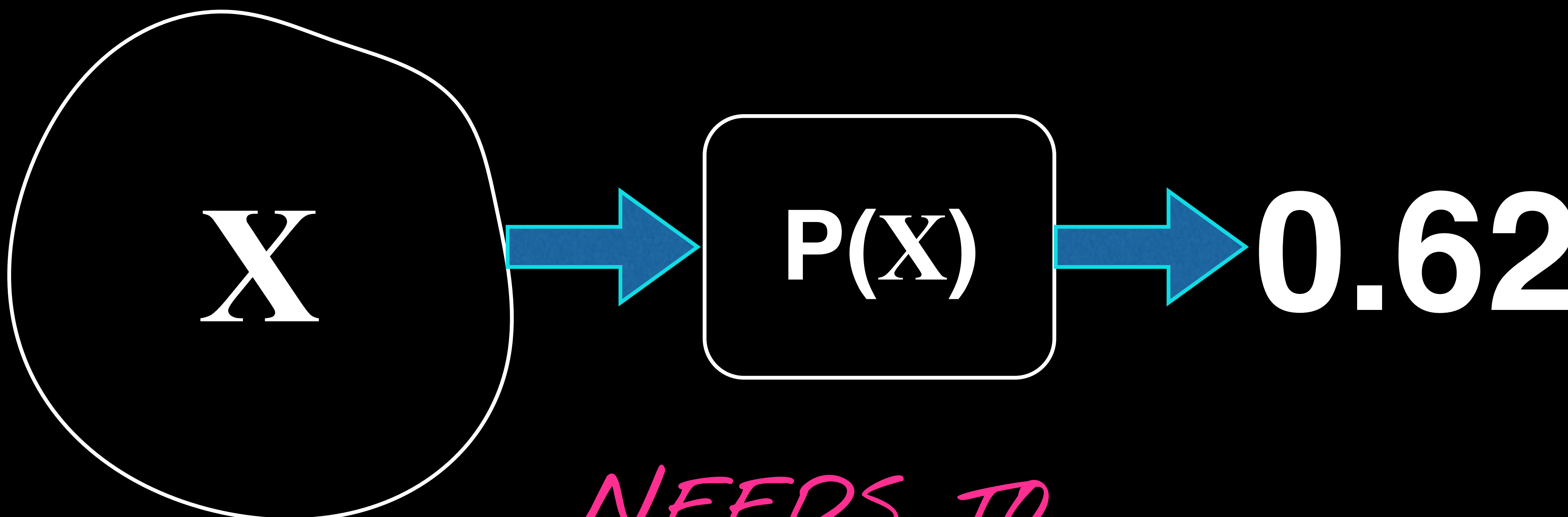
A,B = "odd number and 5" 1/6

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

$$= \frac{\quad}{\quad} = \mathbf{1/3}$$

Bocconi

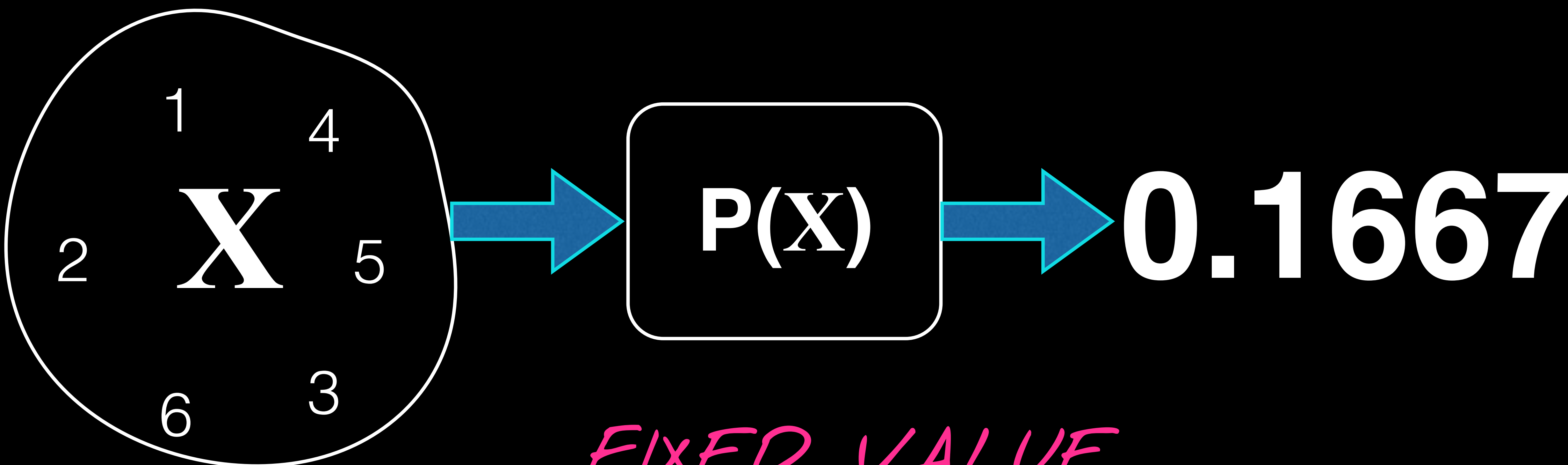
Probability functions



NEEDS TO

- COVER ALL OF X
- SUM TO 1.0

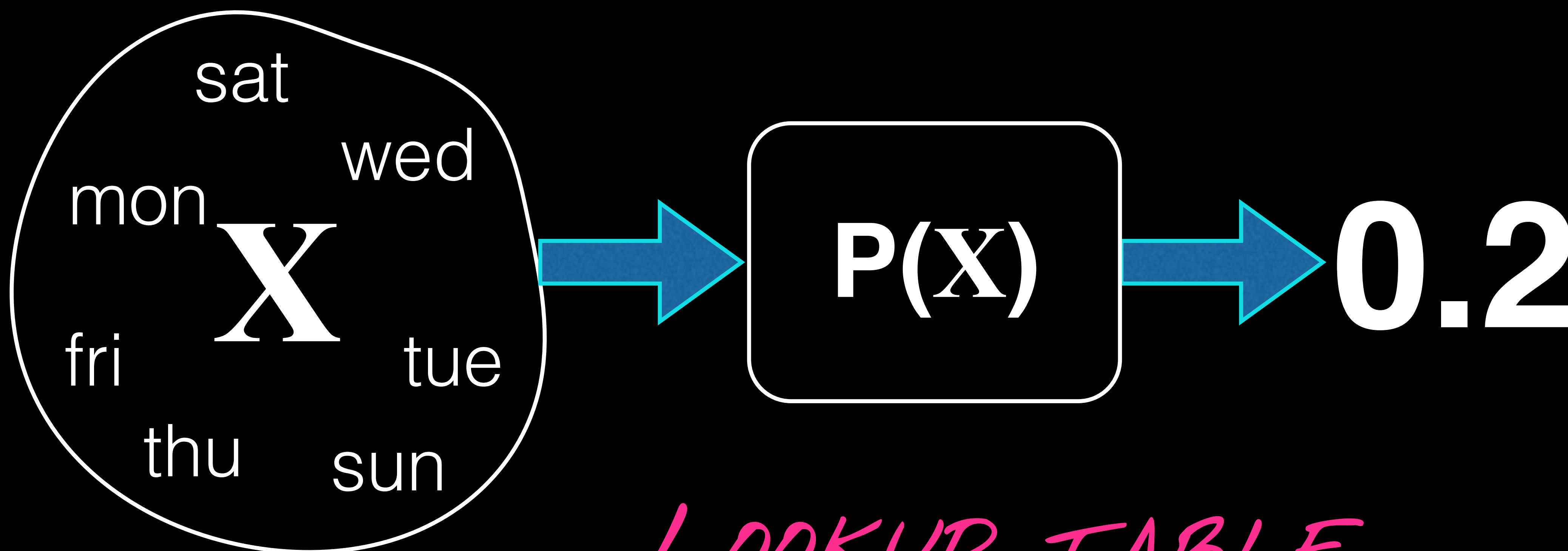
Probability functions



FIXED VALUE

```
def p(number_on_die):
    return 0.1667
```

Probability functions



LOOKUP TABLE

```
def p(me_at_work):
   if me_at_work in [“mon”, “tue”, “wed”, “thu”, “fri”]:
      return 0.2
   else:
      return 0.0
```

Probability functions

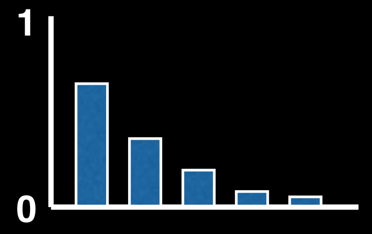
Probability distributions

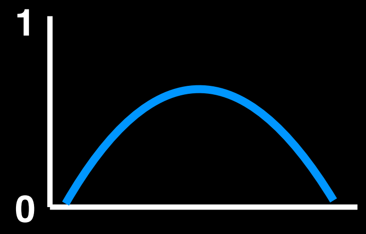
- compute probability for any x
- mathematical way to describe P(x)

```
def p(number_on_die, die_total):
    return number_on_die/die_total
```

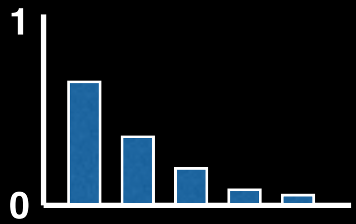
$$P(x; N) = \frac{1}{N}$$

- discrete or continuous





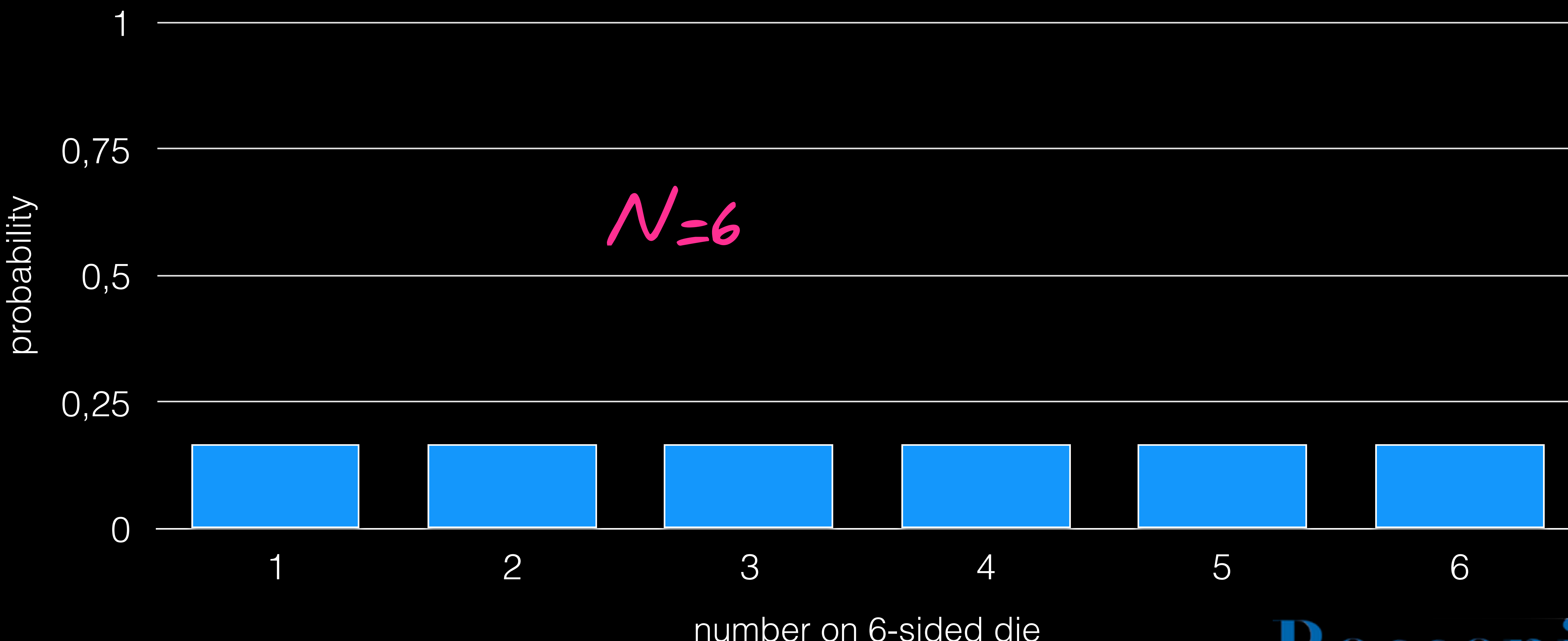
- define “shape” and properties with parameters



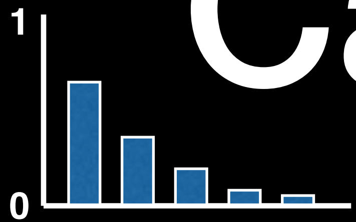
Uniform distribution

Parameters: N NUMBER OF EVENTS

Function: $P(x; N) = \frac{1}{N}$ SUMS TO 1.0



Categorical distribution



Parameters: VECTOR WITH ALL PROBABILITIES

θ

Function: $$P(x;\theta) = \prod_{j=1}^{K} \theta_j^{\mathbb{I}(x_j=1)}$$

USE VALUE AT VECTOR POSITION FOR X

$\theta = [0.33, 0.30, 0.11, 0.07, 0.07, 0.05, 0.04, 0.02]$

"V" $= [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

$P(\text{"V"}) = 0.33^0 * 0.30^1 * 0.11^0 * 0.07^0 * 0.07^0 * 0.05^0 * 0.04^0 * 0.02^0$

Bocconi

Probability and Language

Probability of a Word

"It must be recognized that the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term."

Noam Chomsky

- Choose a word w
- Open a page at random and point at a word:
  Is it w?

HOW OFTEN WE HAVE SEEN W

$$P(w) = \frac{c(w)}{\sum_{v \in V} c(v)}$$

...ALL WORDS

MAXIMUM LIKELIHOOD ESTIMATION

Probability of a Sentence?

HOW OFTEN WE HAVE SEEN SENTENCE S

$$P(S) = \frac{c(S)}{\sum_{z \in Z} c(Z)}$$

...ALL POSSIBLE SENTENCES

$$= \frac{X}{\text{INFINITY}} = 0$$

Can We Make it Simpler?

BREAK IT DOWN!

Andrey Andreyevich Markov
(1856 – 1922)

$$P(S) = P(w_1, w_2, \ldots, w_n)$$

JOINT PROBABILITY OF ALL THE WORDS

Markov Assumption

BREAKING IT DOWN:

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{N} P(w_i | w_1, \ldots, w_{i-1})$$

HISTORY

LIMITING THE HISTORY:

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{N} P(w_i | w_{\underline{i-k}}, \ldots, w_{i-1})$$

Bocconi

Markov Models:

UNIGRAM MODEL (K=0)

$$P(w_1, w_2, \ldots, w_n) \approx \prod_{i=1}^{N} P(w_i)$$

BIGRAM MODEL (K=1)

$$P(w_1, w_2, \ldots, w_n) \approx \prod_{i=1}^{N} P(w_i|w_{i-1})$$

TRIGRAM MODEL (K=2)

$$P(w_1, w_2, \ldots, w_n) \approx \prod_{i=1}^{N} P(w_i|w_{i-2},w_{i-1})$$

Bocconi

A Trigram Model

*** * The weather today is fine STOP**

$$P(S) = P(w_1, \ldots, w_n) = P(\texttt{The} \mid \texttt{* *})$$
$$\times P(\texttt{weather} \mid \texttt{* The})$$
$$\times P(\texttt{today} \mid \texttt{The weather})$$
$$\times P(\texttt{is} \mid \texttt{weather today})$$
$$\times P(\texttt{fine} \mid \texttt{today is})$$
$$\times P(\texttt{STOP} \mid \texttt{is fine})$$

CHAIN RULE

Where Probabilities Come From

`The weather today is …`

WE NEED A WAY TO ASSIGN
P(WORD | "THE WEATHER TODAY IS")

Count in 57m Tweets

MAXIMUM LIKELIHOOD ESTIMATION

(MLE)

```
12 The weather today is just
 9 The weather today is so
 9 the weather today is slightly
 8 The weather today is perfect
 5 The weather today is beautiful
 4 The weather today is slightly
 3 the weather today is so
 3 the weather today is perfect
 3 The weather today is nearly
 3 the weather today is bitter
 3 The weather today is absolutely
 2 The weather today is wonderful
 2 The weather today is beyond
 2 The weather today is amazing
 2 The weather today is a
 1 the weather today is worth
 1 the weather today is weird
 1 The weather today is too
 1 the weather today is the
 1 The weather today is that
 1 the weather today is that
 1 The weather today is splendid
 1 THE WEATHER TODAY IS SO
 1 the weather today is simply
 1 The weather today is sickening
 1 The weather today is seriously
 1 The weather today is pretty
 1 the weather today is pretty
 1 The weather today is Perrfff
 1 the weather today is PERFECT
```

Conditional Probabilities

y	x	P(x|y)*
tea with	milk	0,42
	sugar	0,35
	a	0,18
	stevia	0,05
for the	win	0,25
	majority	0,21
	birds	0,15
	…	…

SUMS TO 1.0

*TOTALLY MADE UP NUMBERS

Bocconi

Smoothing

Many Counts are 0

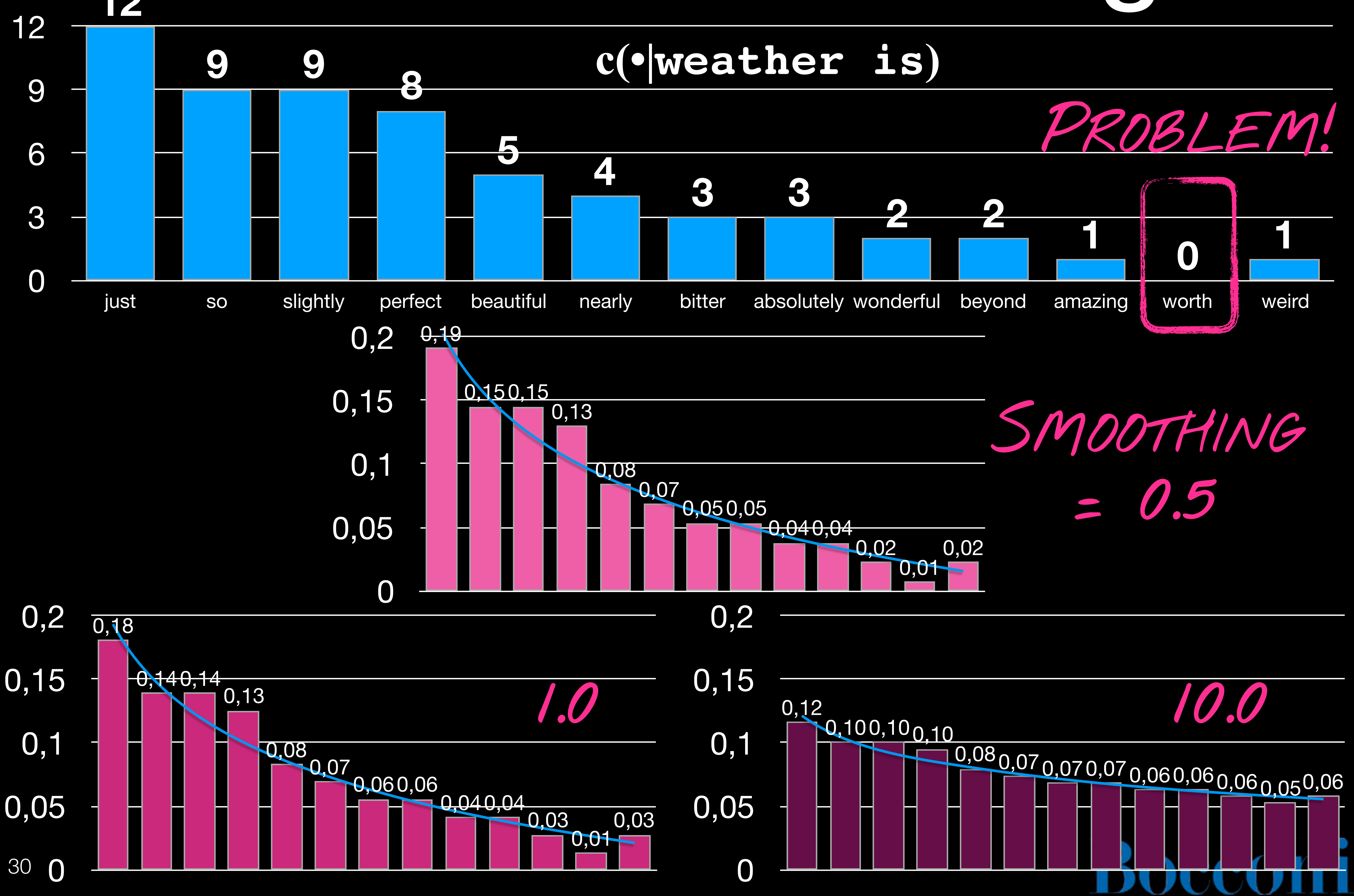
`* * The weather today is fine STOP`

$$P(S) = P(w_1, \ldots, w_n) = P(\texttt{The} | \texttt{* *})$$
$$\times P(\texttt{weather} | \texttt{* The})$$
$$\times P(\texttt{today} | \texttt{The weather})$$
$$\times P(\texttt{is} | \texttt{weather today})$$
$$c(\texttt{fine}) = 0 \qquad \times P(\texttt{fine} | \texttt{today is})$$
$$\times P(\texttt{STOP} | \texttt{is fine})$$

Add-one (Laplace) smoothing

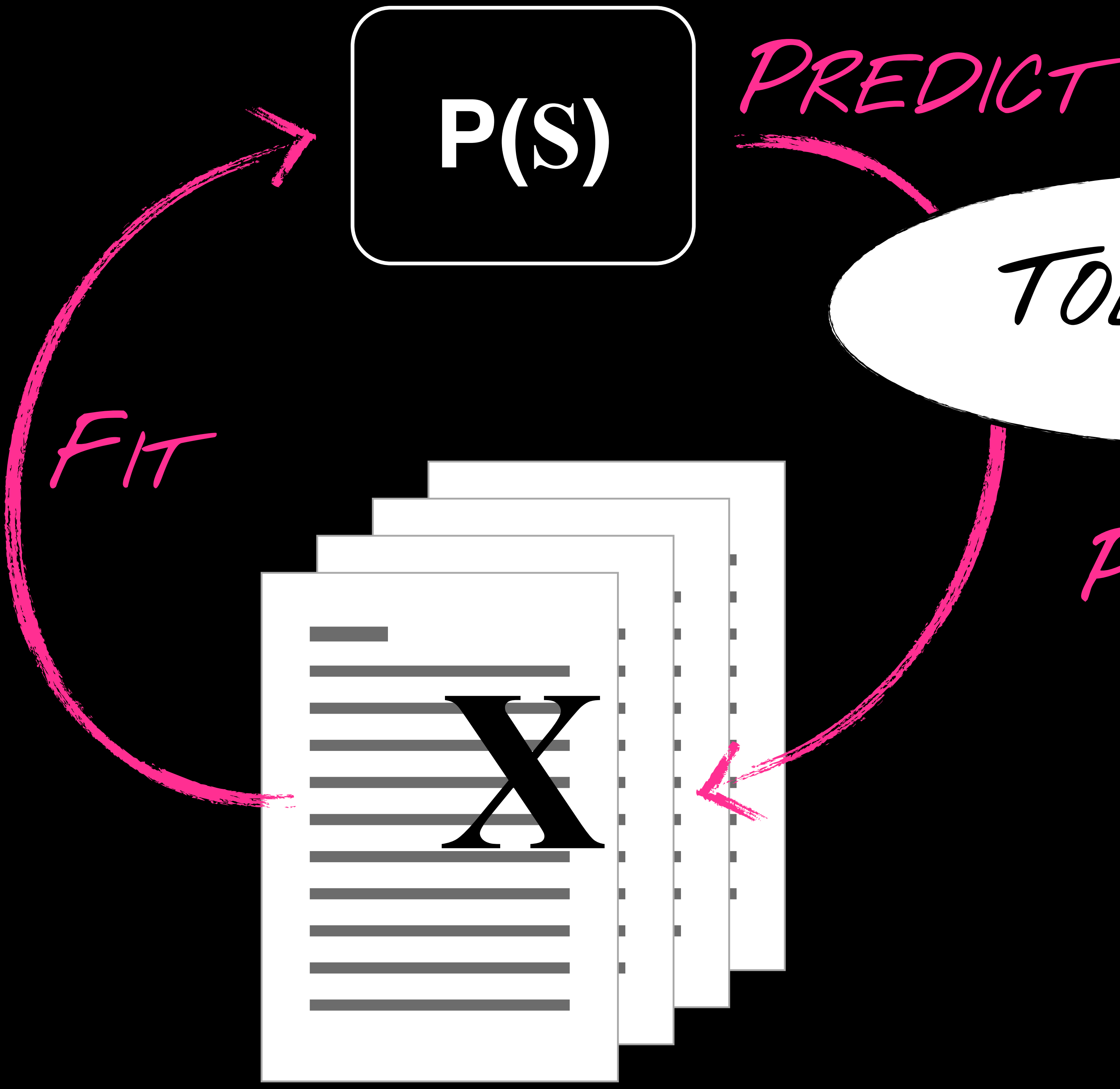
Pierre-Simon, marquis de Laplace
(1749 – 1827)

Effect of Smoothing



Evaluating LMs

How Good is My Model?



PERPLEXITY $2^{H(X)}$

$$= 2^{-\sum_x p(x) \log p(x)}$$

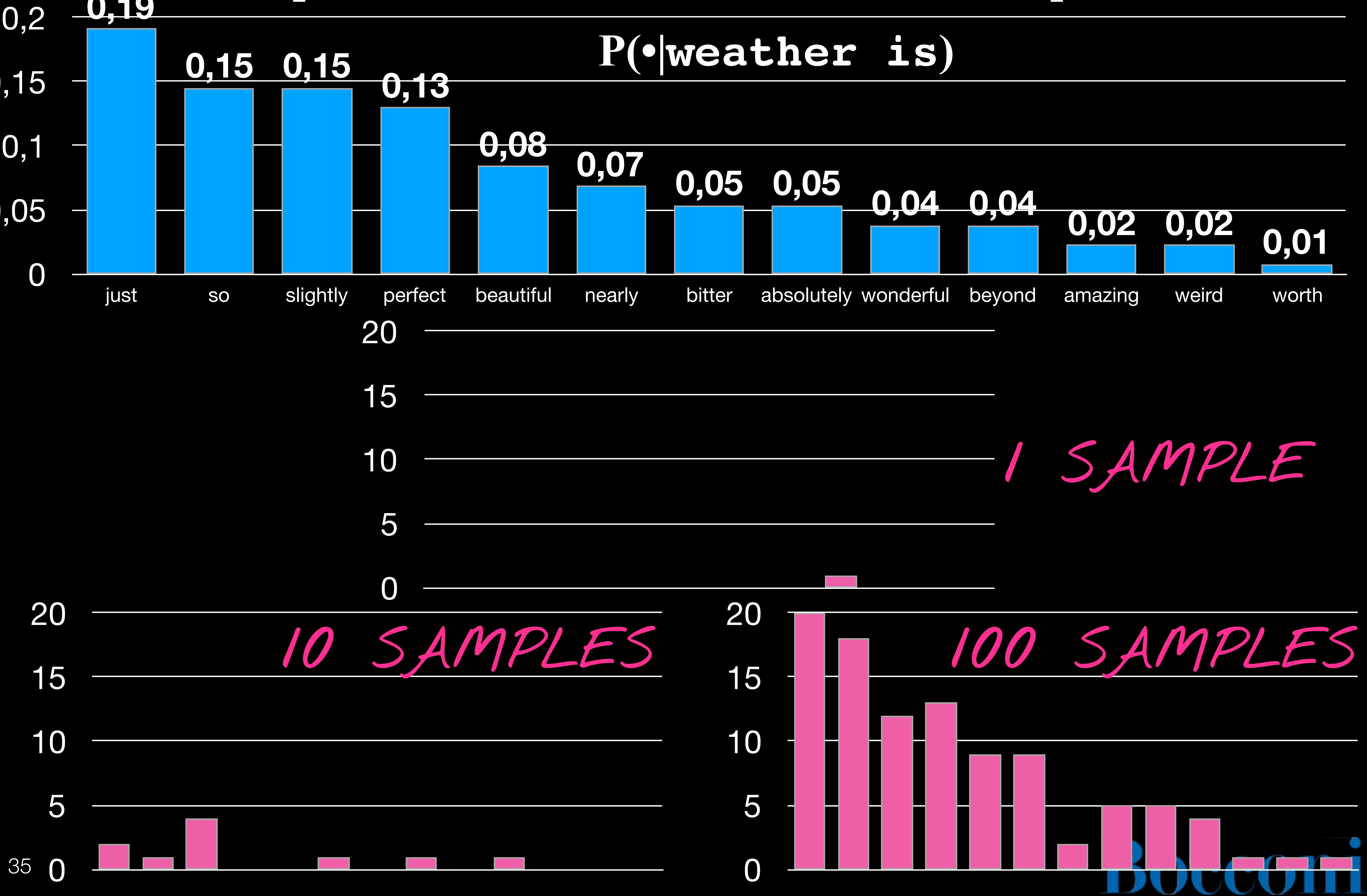
Using LMs for Generation

Take a Random Walk

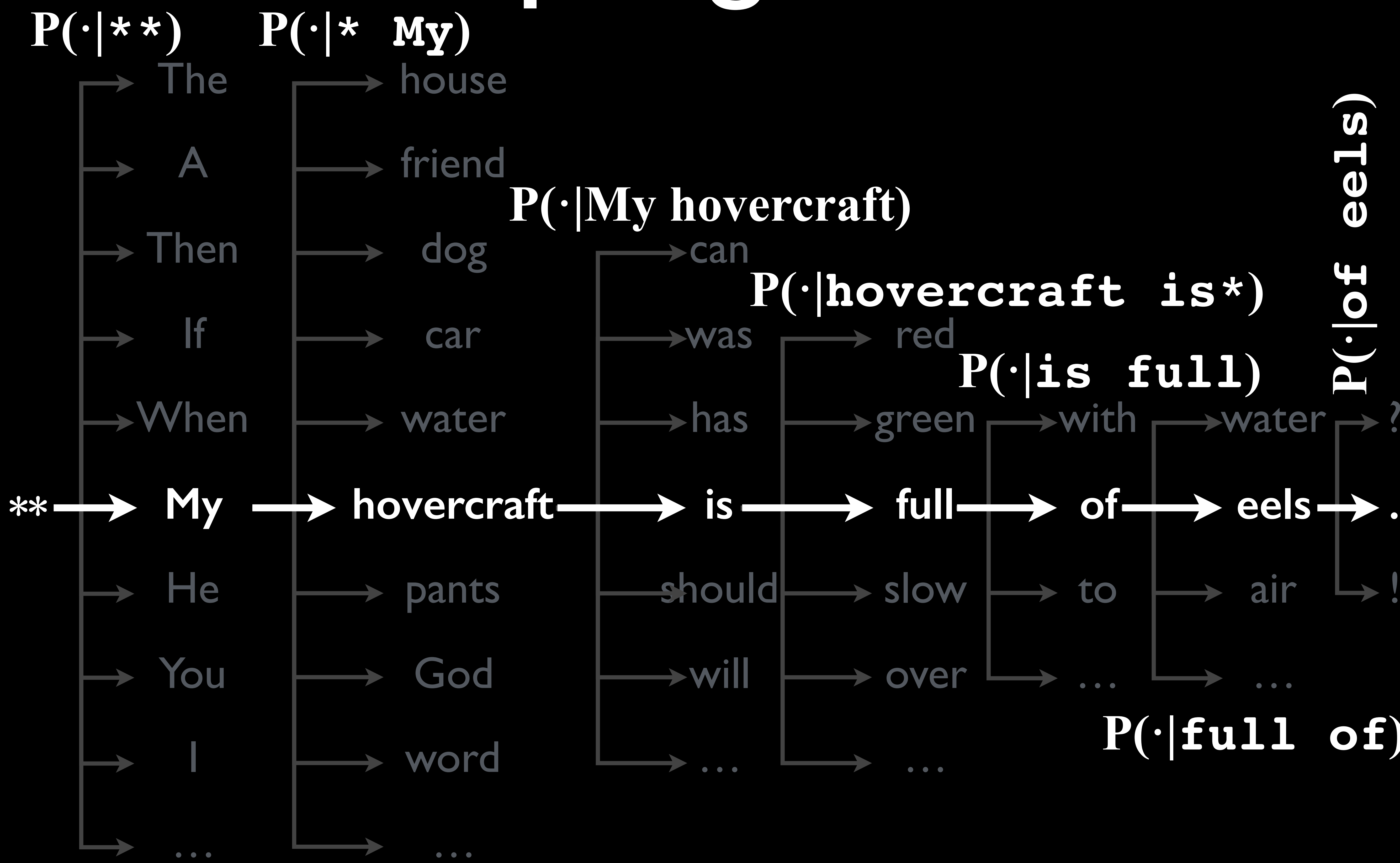
PROPORTIONATELY

```
Pick a random word w from P(·| * * )

H = [*, *, w]

While H[-1] is not STOP:

    Pick a random word w from P(·|H[-2:])

    H += [w]

return H
```

Proportionate Samples



Sampling Words



Wrapping Up

Language Model in Short

1. Break sentence into *n*-grams MARKOV HORIZON
2. Increase their counts SMOOTHING
3. Compute probabilities MLE
4. Multiply them together MARKOV ASSUMPTION

Bocconi

Take-Home Points

- **Language Models** assign a probability to any sentence
- The **Markov assumption** breaks sentence probability into a **chain** of word conditional probabilities
- **Markov order** determines the size of the conditional *n*-grams
- *n*-gram probabilities can be computed with **Maximum Likelihood Estimation** from a large corpus
- **Smoothing** helps address the problem of unseen words
- The same LM parameters can be used for **text generation**