Grokking Anchors:

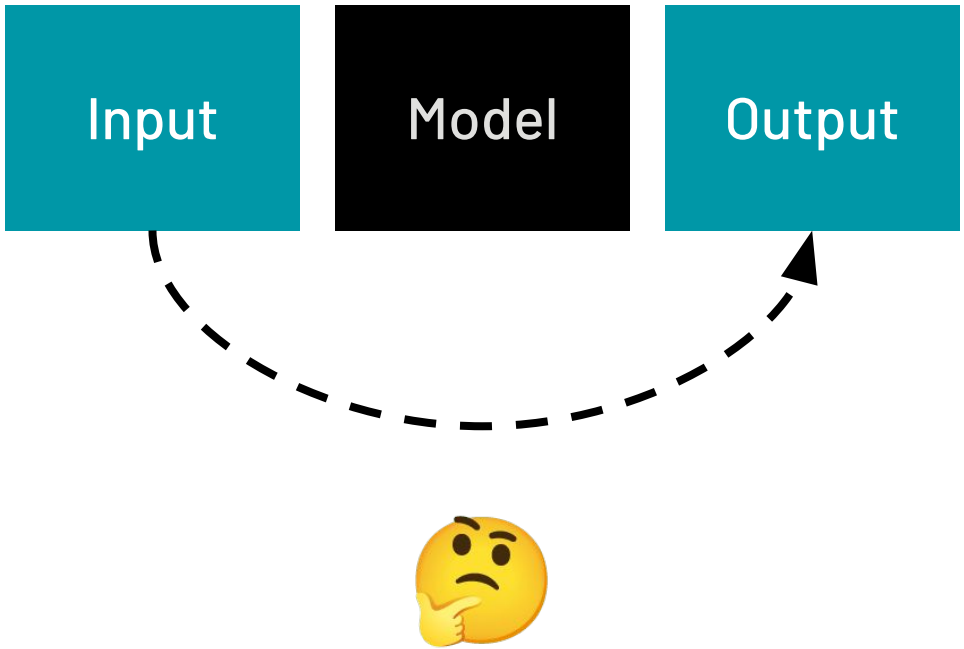
Uncovering what a machine-learning model relies on

Kilian Kluge | PyConDE & PyData Berlin 2023 | April 19th 2023

inlinity
trusting artificial intelligence

XAI Studio

Erklärbare Künstliche Intelligenz

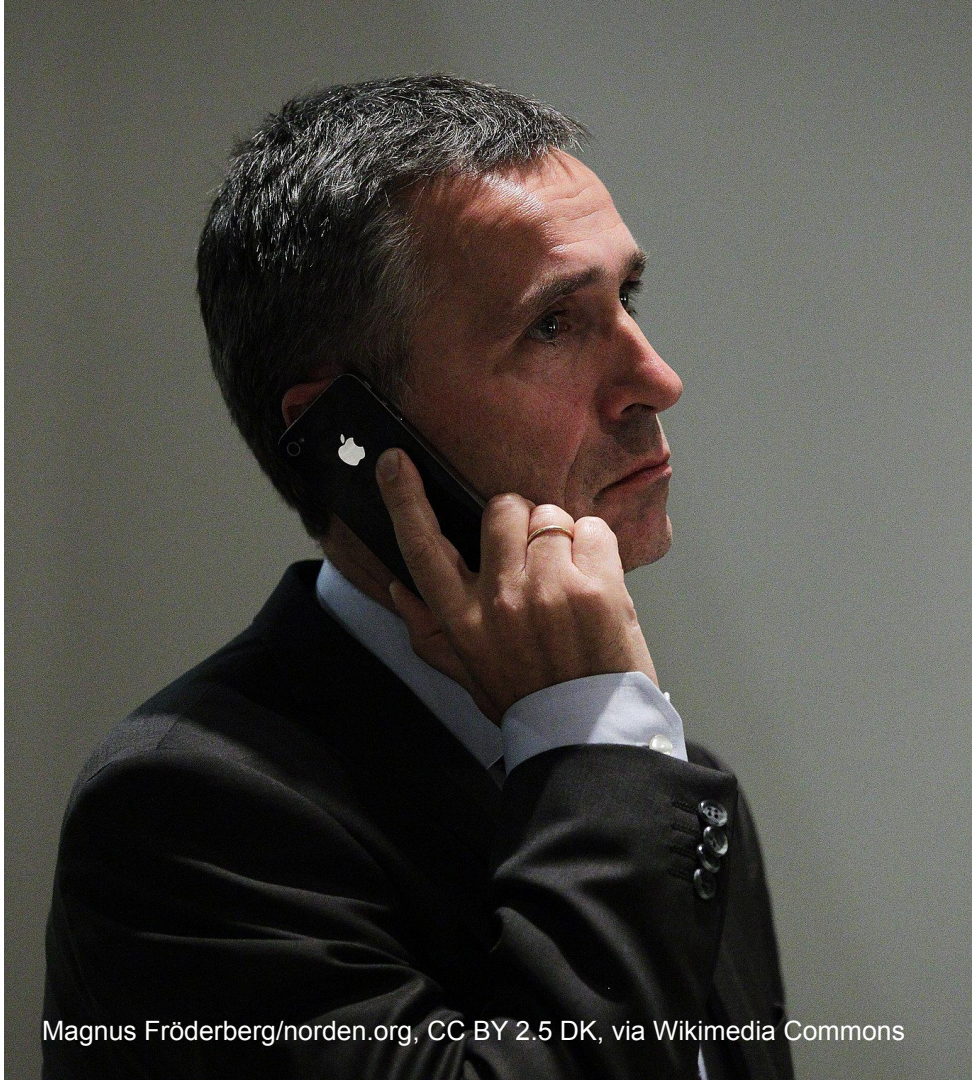

Input
Model
Output

Roadmap

1) What is an “anchor”?
2) Defining anchors as rule sets
3) Basic strategy for finding anchors
4) KL-LUCB multi-armed bandit algorithm
5) Assembling a complete anchor algorithm

What is an “anchor”?

Silvio John, Elsterwerda, CC BY-SA 2.5, via Wikimedia Commons



Magnus Fröderberg/norden.org, CC BY 2.5 DK, via Wikimedia Commons

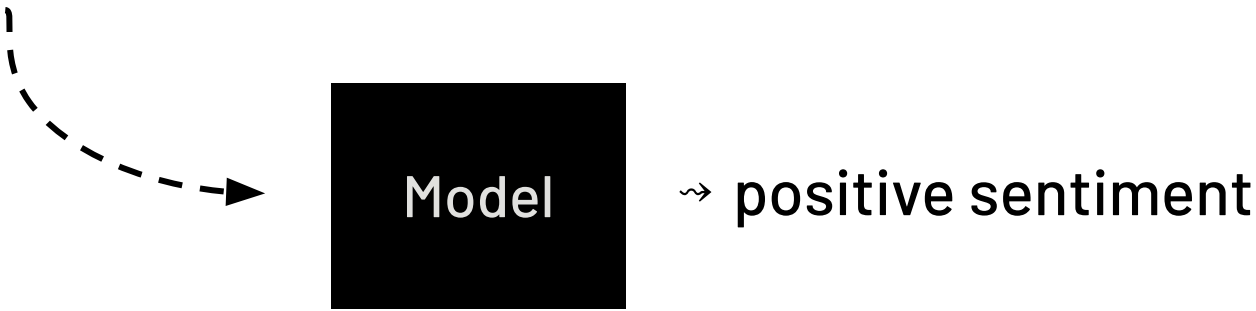
As long as the anchor is present,
we maintain our decision.

Bert Kaufmann from Roermond, Netherlands, CC BY-SA 2.0, via Wikimedia Commons

As long as the anchor is present
in the input, the model maintains its output.

Today's example

"Python is the best programming language."



Defining anchors as rule sets

Let’s create some rules!

Python	is	the	best	programming	language

Does the sentence contain “Python”?

Does the sentence contain “is”?

Does the sentence contain “the” ?

Does the sentence contain “best”?

Does the sentence contain “programming”?

Does the sentence contain “language”?

Binary rules

Any rule that can be encoded as **`True`**/**`False`**.

Examples:

- Presence of a token or word
- Presence of a pixel or image segment
- A feature's value surpasses a given threshold

Complex rules

Any rule that requires more than one bit to encode.

Examples:

- A lower and upper bound on a scalar feature's value
- A set of possible values for a categorical feature

An anchor is a set of rules.
As long as all rules are fulfilled
for the input, the model maintains its output.

Our first (trivial) anchor

`True` Does the sentence contain “Python”?

`True` Does the sentence contain “is”?

`True` Does the sentence contain “the” ?

`True` Does the sentence contain “best”?

`True` Does the sentence contain “programming”?

`True` Does the sentence contain “language”?

What is a good anchor?

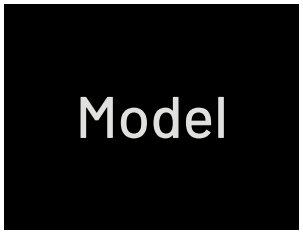
We’re looking for **the smallest set of rules** such that

set of rules applies to input ⇒ model yields original output

holds with high probability

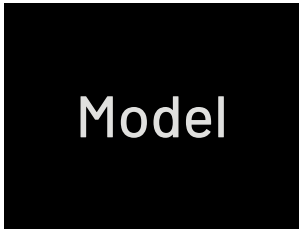
What's our assumption?

"Python is the best programming language."



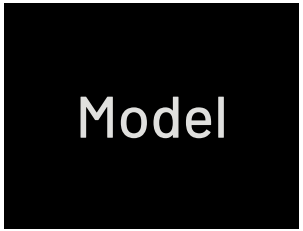
⇝ positive

"Java is the best programming language."



⇝ positive

"Python is the worst programming language."



⇝ negative

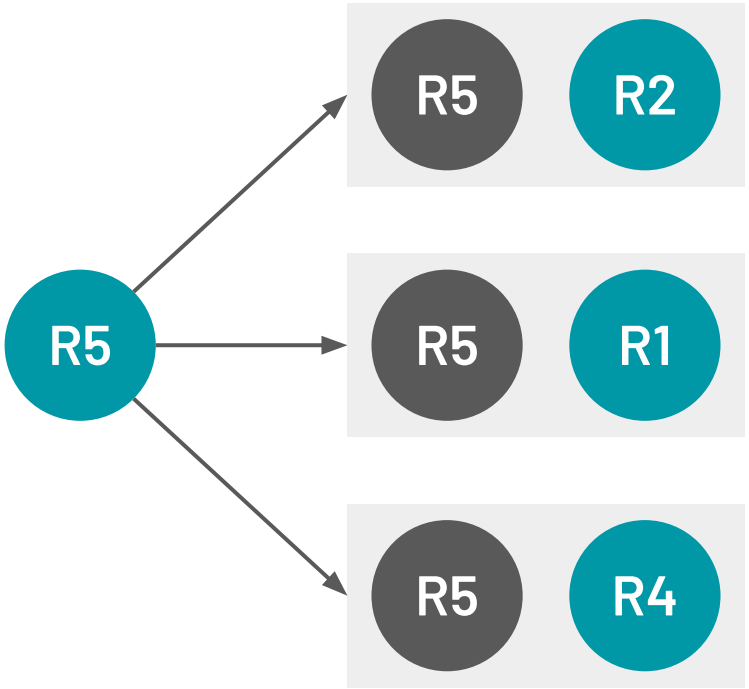
Basic strategy for finding anchors

Local beam search

R5

Does the sentence contain “programming”?

Local beam search

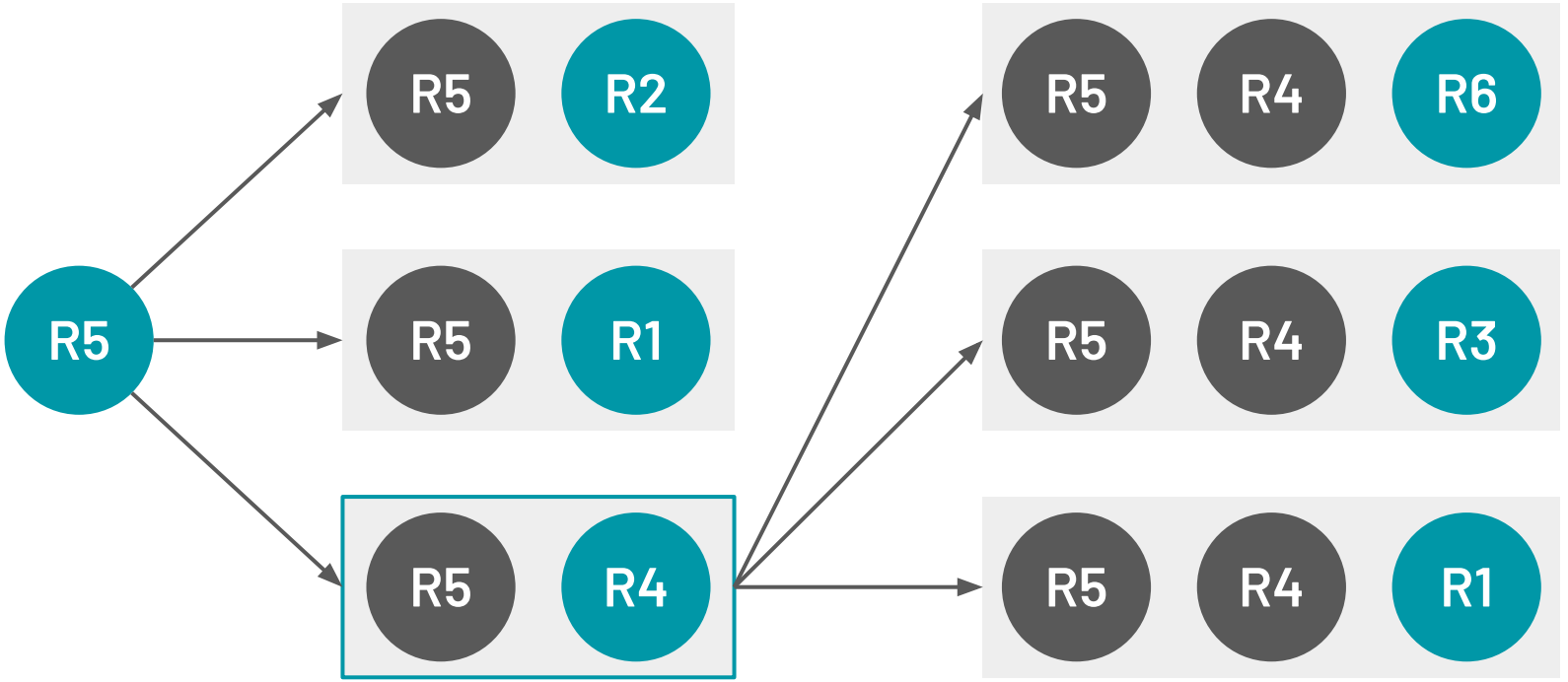


Does the sentence contain “programming” and “is”?

Does the sentence contain “programming” and “Python”?

Does the sentence contain “programming” and “best”?

Local beam search



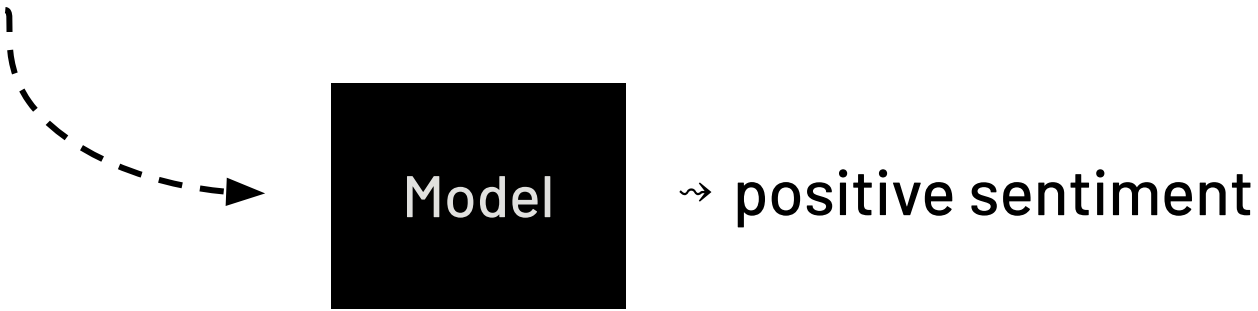
Evaluating anchor candidates (I)

Sentence contains the words "programming" and "best"

"Python is the worst programming language."	**`False`**
"Python is the worst programming tool."	**`False`**
"Java is the worst programming language."	**`False`**
"Java is the best programming language."	**`True`**

Evaluating anchor candidates (II)

“Java is the best programming language.”



Multi-armed bandits 101

Beliefs

R5 R2	*P(R5 & R2 ⇒ positive sentiment)*
R5 R1	*P(R5 & R1 ⇒ positive sentiment)*
R5 R4	*P(R5 & R4 ⇒ positive sentiment)*

"pull"
&
"update"

KL-LUCB multi-armed bandit algorithm

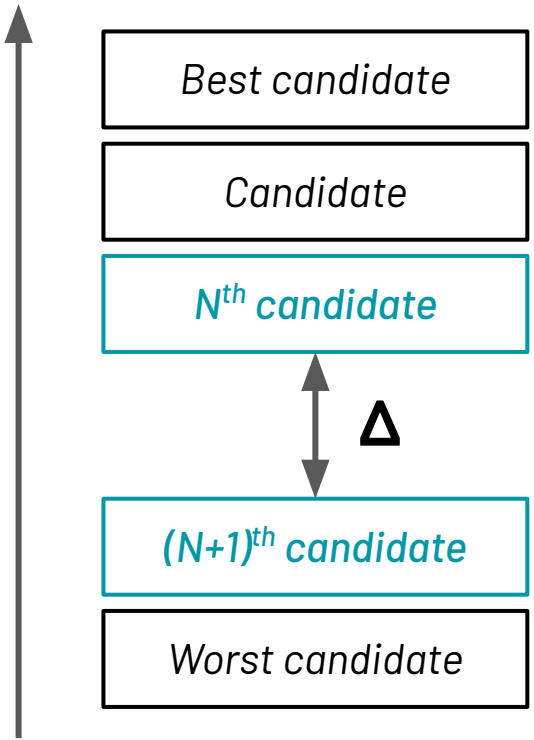
KL = Kullback-Leibler divergence bounds

We sampled an anchor candidate **N** times.

We obtained the desired output in **M** cases.

What are the lowest and highest “true probabilities” for
set of rules applies to input ⇒ model yields original output
that are compatible with this observation?

LUCB = Lower/Upper Confidence Bound



We want to find the best N candidates.

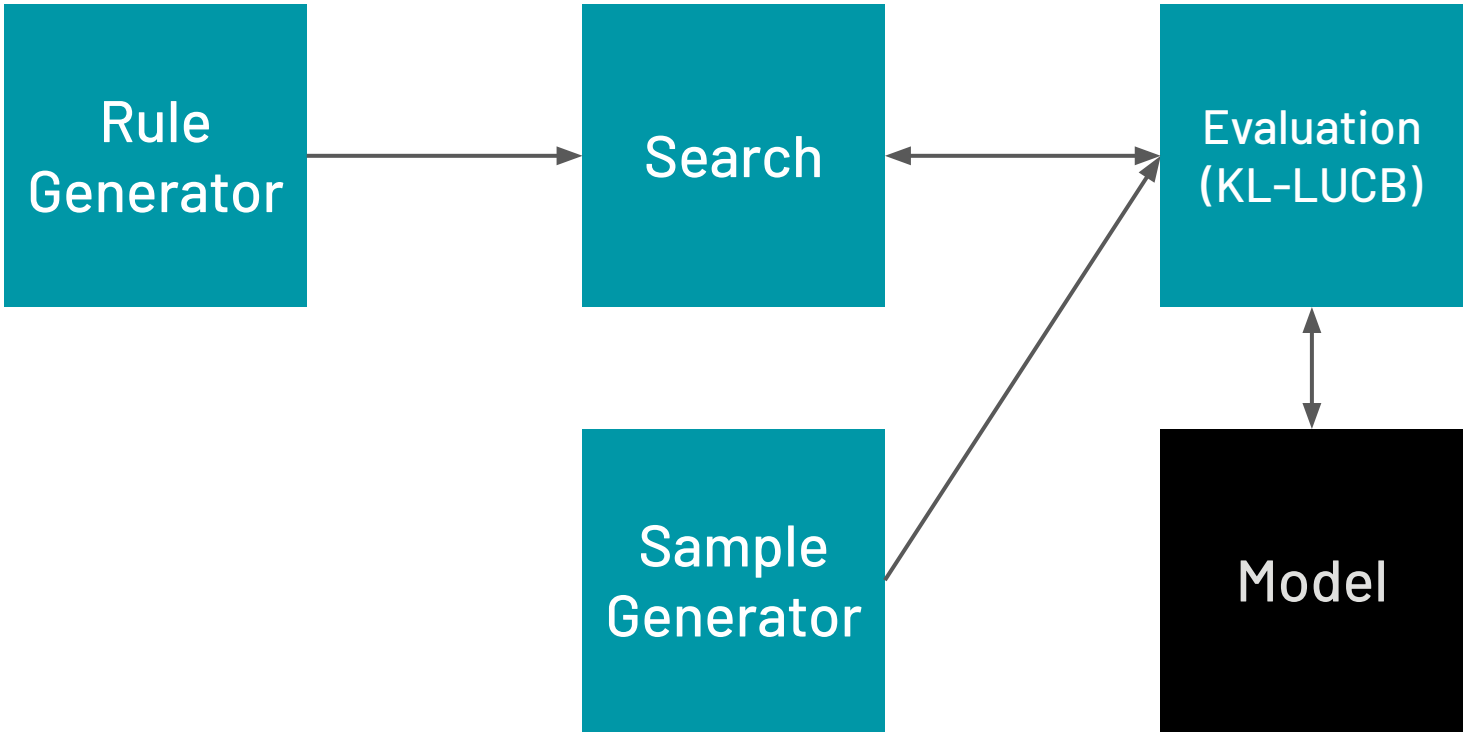
1) Sort the candidates based on the current beliefs

2) **Refine our beliefs about the N^{th} and the $(N+1)^{th}$ candidate**

Repeat until the **lower bound for the N^{th} candidate and the upper bound for the $(N+1)^{th}$ candidate differ by Δ**

Assembling an anchor algorithm

Main components of an anchor implementation



Thank you!

github.com/ionicsolutions/grokking-anchors