Searching for Model Concepts by their Conditional Independencies

TBD

Abstract

Your abstract here

Ongoing attributions:

- David: articulating connection to his explainability agenda. Writing first draft.
- Kaarel: suggesting the search algorithm
- Victor: suggesting obstacles related to rotations
- AIs: ChatGPT and Copilot

1 Motivation

A lot of human understandable concepts have statistical independencies (e.g. because of underlying causal processes). These conditional independencies should help us narrow the search for concepts in a model, in the spirit of the "Discovering Latent Knowledge" paper. [todo: cite]

This research direction is insturmentally valuable to David, because he's focused on context-specific, model-agnostic explainability. But the direction he's going in relies on at least some of the relevant concepts being readable from the model, so if there are easy ways he can help speed that interpretability research, it'll make his particular agenda of explainability more likely to succeed.

2 Formalizing the problem

Assume we have three binary concepts X, Y, and Z which satisfy the conditional independence of $(X \perp\!\!\!\perp Y|Z)$, and we want to find them in a transformer-based generative model. We can do this by searching for a set of three features in the model which satisfy the conditional independence.

2.1 What are we searching over?

Let's assume concepts are represented linearly as subspaces of the embedding space, and the i^{th} layer is simply reasoning over these concepts (but not changing the embedding space itself). If this is true, **the way that concepts are represented should be consistent in the embedding space across all layers**.

We can think of the overall search as comprising two parts:

- Searching over all possible combinations of three subspaces in the embedding space.
- For each combination, checking whether the conditional independence holds.

2.2 Do we need to worry about rotations?

Another relevant question is whether there is a privileged basis in the embedding space. If there is, then we can search over the basis vectors. If there isn't, then we need to search over all possible rotations of the embedding space.

cite: https://transformer-circuits.pub/2023/privileg basis/index.html

Anthropic's work has provided a bit of evidence that there **shouldn't be a priveledged basis**. Empirically, there was some evidence that there is anyways, but Anthropic ran some tests and conjecture that it's likely just an artifact of using the Adam optimizer, and not necessary for model performance. So at a first pass, it would seem we need to search over all possible rotations of the embedding space.

If we restrict our search to only the columns of the interaction matrices, what assumptions are we making?

- There is a one-to-one correspondence between concepts and columns of the interaction matrices. (In particular, there are no concept-directions which are linear combinations of the columns of the interaction matrices, and *superposition* isn't a thing).
- there are no concept-directions corresponding to interactions between layers (read-write operations of later layers building on the read-write operations of earlier layers)

I think the superposition point is the biggest reason we're going to need to worry about rotations. If we assume that superposition is a thing, then we can't assume that the columns of the interaction matrices are concept directions.

2.3 How do we check conditional independencies?

Regardless of how we enumerate features, we need to be able to check whether the conditional independencies hold. Naively, we could just check whether the conditional independencies hold for a triplet of features. This is doable for discrete features, but challenging for continuous features [todo: verify].

It would be nicer if we could specify a loss function which is minimized when the conditional independencies hold. This would allow us to use gradient descent to find the features which satisfy the conditional independencies.

For instance, perhaps we could use a loss function based on KL divergence between the product distribution and the joint distribution on that batch.

Maybe specifying a continous loss function also allows us to get around the rotation issue, because the rotation is just another set of parameters to do gradient descent over? So long as it doesn't privledge any particular basis?

2.4 Differentiating relevant concepts

Any set of three concepts which satisfy the conditional independencies will be a valid solution to the search problem. So for the type of application David is imagining (finding concepts which are relevant to a particular context/task), we need a way to ensure the concepts we find are relevant to the context/task.

Maybe this can be added to the loss function somehow? For instance, maybe we can add a term to the loss function which is minimized when the concepts are relevant to the context/task (aka. supervised probing).

2.5 Narrowing the search

Perhaps trying to satisfy several conditional independencies simultaneously helps narrow the search. Maybe searching for 5 concepts with 4 conditional independencies is easier than

searching for 3 concepts with 1 conditional independence?

Alternatively, since LLM's seem to be capable of answering questions about the conditional independnece of concepts, maybe we can leverage their own understanding somehow? This would indicate that they have some sort of representation of conditional independencies, which we could use to narrow the search.

A April 25 Victor summary

- Assume you have infinite compute and infinite data
- It seems like even then you may not be able to do this search easily, because there are infinite rotations you would need to check (there's no privileged basis).
- The counterpoint is that LLMs seem to be able to also have an understanding of which concepts are conditionally independent: if you assume that knowledge is also represented somewhere in the model, does that buy you the extra constraints you need to narrow down the search space to something that ever terminates?

B April 25 discussion with Kaarel

- Transformer model
- Causal graph has 3 nodes (actually a CI)
- Try all possible variations of 3 places in the residual stream to look at
  - Both token position and layer: all possible combinations of places
- Look for a feature in each position, testing whether each feature has the independence you want
- Should be possible to do using a differentiable loss function
  - If binary concepts are sigmoids of affine transformations of features
  - Then loss depends on whether the CI holds in that batch
  - KL divergence between product distribution and joint distribution you observe
  - Need an empirical variant of the mutual information.
- Gradient descent is on how to choose 3 positions in the model.
- If there are any other concepts which have the same causal structure, this won't be able to differentiate between them
  - So we could add a supervised term which matches prompt = "man"
- If the 3 variables have deterministic value from a given input, then we could train a supervised probe to match the labels I know to the positions in the model.