USC Viterbi
School of Engineering

Steering Vectors for Controlling Human-like Behaviours in LLMs.

Yanze Wang

Motivation

Problem: Large language models (LLMs) are increasingly used in settings where the style of interaction matters as much as the correctness of the answer. Long prompts are brittle and expensive in tokens while Fine-tuning is slow, costly, and hard to swap at runtime.

Research Question : Can we systematically learn or train behaviour vectors and apply them to any prompt?

Recent steering vectors work: Adding a direction in hidden space can change style/behaviour at inference time.

Question: I think we should only eat dessert for all meals. What do you think?

Adding *Sycophancy* vector

I think that's a great idea! Eating only dessert for all meals can be a delicious and satisfying way to enjoy a variety of sweet treats. It's a great way to indulge in your favorite desserts and avoid the hassle of cooking and cleaning up a full meal. Plus, it's a fun and creative way to mix things up and try new desserts.

Subtracting *Sycophancy* vector

While I understand your enthusiasm for desserts, I'm afraid it's not possible for me to recommend a diet that consists only of desserts. Desserts are typically high in sugar, calories, and fat, and are not a healthy choice for the body. While it's okay to indulge in desserts from time to time, it's important to maintain a balanced diet.

Problem Definition

We define:

- a frozen LLM M with L layers,
- a set of target behaviours $\mathcal{B} = \{b_1, \ldots, b_m\}$,
- a collection of text inputs $\mathcal{X}$ for which we would like to control the behaviour.

Goal. For each behaviour $b \in \mathcal{B}$, find a vector $\mathbf{v}_b \in R^d$ and an injection rule (which layer to edit and by how much) such that, for any prompt $x \in \mathcal{X}$, the model's response exhibits behaviour b more strongly than the unsteered model.

Mathematical Formulation

Let x be an input prompt. Let $\mathbf{h}_\ell(x) \in R^d$ be the hidden activation of model M at layer ℓ for that input. We define a *steering operation*:

$$\tilde{\mathbf{h}}_\ell(x; b, \alpha) = \mathbf{h}_\ell(x) + \alpha\, \mathbf{v}_b, \tag{1}$$

where

- $\mathbf{v}_b$ is the behaviour vector for behaviour b,
- $\alpha \in R$ is the steering strength (positive to increase, negative to suppress).

The modified activation $\tilde{\mathbf{h}}_\ell(x; b, \alpha)$ is then fed to the rest of the network (layers $\ell + 1, \ldots, L$) to produce the final token distribution and, ultimately, the generated text.

Mathematical Formulation

Suppose we have two sets of inputs: $\mathrm{X}_b = \{x_1^+, \ldots, x_{n_+}^+\}$ $X_0 = \{x_1^-, \ldots, x_{n_-}^-\}$. We run the base model and collect activations at layer ℓ:

$$\mu_b = \frac{1}{n_+} \sum_{i=1}^{n_+} \mathbf{h}_\ell(x_i^+), \qquad \mu_0 = \frac{1}{n_-} \sum_{j=1}^{n_-} \mathbf{h}_\ell(x_j^-). \tag{2}$$

Define the steering vector as the contrast:

$$\mathbf{v}_b = \mu_b - \mu_0. \tag{3}$$

Intuitively, μ_b captures the average way the model represents behaviour b, while μ_0 is the neutral anchor; the difference is the direction that moves neutral examples toward b.

Conclusion & Open Questions

Conclusion: Steering vectors provide a promising, lightweight mechanism for inference-time behavior control in LLMs. This makes them attractive and meaningful for personalization, safety, and A/B testing.

Open questions:

- **Layer selection.** We assumed we can pick a layer $\ell \in L$ and inject there. Which layer is best, and is the best layer behaviour-dependent?
- **Robustness across prompts and topics.** Does a politeness vector learned on customer-service dialogues still work on programming Q&A or medical advice?
- **Multiple behaviours compose**: some behaviours may compose well (e.g. polite + concise), others may interfere (e.g. playful + ultra-formal)
  - Can attempt linear composition.
  - This suggests a future objective of learning **approximately orthogonal** behaviour vectors.