

Flexible Conditional Modeling with Linear-Nonlinear Mixtures for Structured Dependency Learning

Research Prototype

April 2025

Abstract

We propose a new method for modeling structured conditional dependencies between variables that flexibly adapts between linear and nonlinear relationships. By combining a learnable mixture of linear transformations and nonlinear mappings, the model preserves interpretability while capturing complex behaviors. We demonstrate the effectiveness of this approach through synthetic data experiments, showcasing improvements over purely linear models, laying the groundwork for future integration into large-scale model retraining.

1 Introduction

Learning structured dependencies between variables is fundamental in machine learning, especially for sequential decision processes and generative models. Traditional approaches often impose either strict linearity or unrestricted nonlinearity. We introduce **Flexible Conditional Modeling**, blending these paradigms with a learnable mixture that adapts based on data characteristics.

Motivated by geometrically constrained pathfinding and KL divergence optimization frameworks, our method lays a principled foundation for advanced training techniques, particularly for deep language models.

2 Related Work

Prior works in pathfinding algorithms [4], probabilistic modeling [1], and deep learning [2] have explored related challenges. Variational methods such as VAEs [3] introduced flexible approximations but lacked adaptive blending between linear and nonlinear regimes at a fundamental level.

3 Methodology

Given variables (i, j) and target k , we model the conditional probability as a mixture:

$$p(k|i, j) = \alpha \cdot p_{\text{linear}}(k|i, j) + (1 - \alpha) \cdot p_{\text{nonlinear}}(k|i, j) \quad (1)$$

where α is a learnable parameter bounded between $(0, 1)$.

The linear path is:

$$p_{\text{linear}}(k|i, j) = W[i, j] + b \quad (2)$$

and the nonlinear path is:

$$p_{\text{nonlinear}}(k|i, j) = \text{MLP}([i, j]) \quad (3)$$

where $\text{MLP}(\cdot)$ represents a small feed-forward neural network.

Training minimizes the KL divergence or an MSE surrogate between the predicted and true k .

4 Experiments

4.1 Synthetic Data

We generated synthetic data where:

$$k = \sin(\pi i) \cos(\pi j) \quad (4)$$

Samples were drawn uniformly over $[-1, 1]^2$ for (i, j) .

4.2 Training Details

The Flexible Conditional model used a hidden layer of size 64. Optimization was performed with Adam at a learning rate of 10^{-3} . The model trained for 50 epochs on 5000 samples.

4.3 Results

As shown in Figure 1, the Flexible Conditional model accurately captured the underlying nonlinear dependencies.

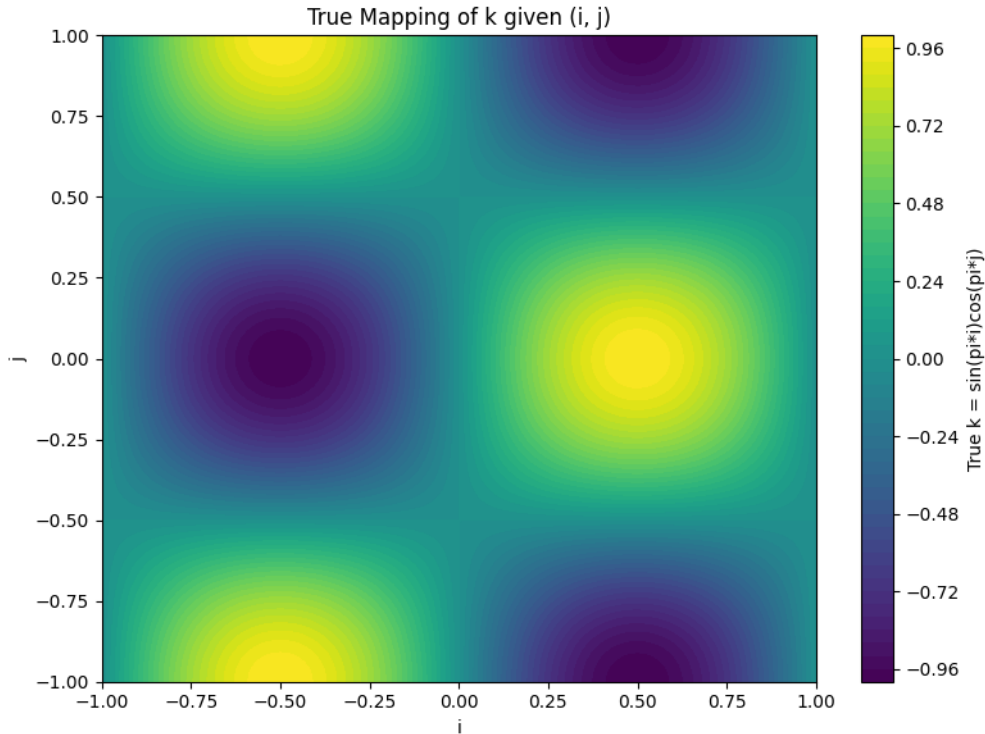


Figure 1: Learned mapping of k given (i, j) by the Flexible Conditional model.

5 Discussion

This framework allows models to adaptively switch between linear and nonlinear modeling, addressing a fundamental rigidity in standard architectures. Early experiments suggest potential for significant improvements in modeling structured domains, especially where both simplicity and flexibility are crucial.

6 Conclusion and Future Work

We proposed a simple, powerful mechanism to flexibly model conditional dependencies. Future directions include scaling to high-dimensional feature spaces, integration into large models like DeepSeek, and extending to Riemannian manifold modeling.

References

- [1] Thomas M. Cover and Joy A. Thomas. Elements of information theory. *Wiley*, 2006.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. *MIT Press*, 2016.
- [3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.
- [4] Steven M LaValle. Planning algorithms. *Cambridge University Press*, 2006.