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

Research Prototype

April 2025

Abstract

We propose a flexible conditional modeling framework that learns structured dependencies between variables by blending linear and nonlinear transformations. Our method dynamically adjusts to data characteristics via a learnable mixture parameter, allowing the model to preserve simplicity where appropriate while capturing complex behaviors when necessary. We validate the approach on both simple synthetic nonlinear datasets and more challenging structured datasets, achieving competitive or superior performance compared to traditional linear models.

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:

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

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

The linear path is:

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

and the nonlinear path is:

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

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 Simple Synthetic Data

We generated synthetic data where:

$$k = \sin(\pi i)\cos(\pi j) \tag{4}$$

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 5×10^{-4}. The model trained for up to 1000 epochs on 5000 samples.

4.3 Spiral Dataset

To further evaluate our model under more challenging conditions, we generated a two-dimensional spiral dataset. Data points (x, y) were distributed along spiraling curves, and the target value k was computed as:

$$k = \sin(0.5\sqrt{x^2 + y^2}) \tag{5}$$

This creates a strongly nonlinear structured task where traditional MLPs tend to struggle.

4.4 Results

Initially, the standard MLP achieved faster convergence and slightly lower final loss on simple synthetic tasks. However, given longer training (up to 1000 epochs), the Flexible Conditional model matched or slightly surpassed the performance of the standard MLP. On the more challenging spiral dataset, the Flexible Conditional model demonstrated superior generalization and adaptability, as visualized by a smoother loss curve and better prediction alignment.

5 Discussion

This framework allows models to adaptively switch between linear and nonlinear modeling, addressing a fundamental rigidity in standard architectures. While simple tasks may not showcase its full strength, our experiments on structured, curved data demonstrate its ability to adapt and outperform traditional methods. Longer training periods help the flexible model better optimize its internal mixture parameter.

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, expanding evaluations to additional structured datasets (e.g., circular arcs, manifold paths), tracking mixture parameter evolution during training, and integrating into large models like DeepSeek.

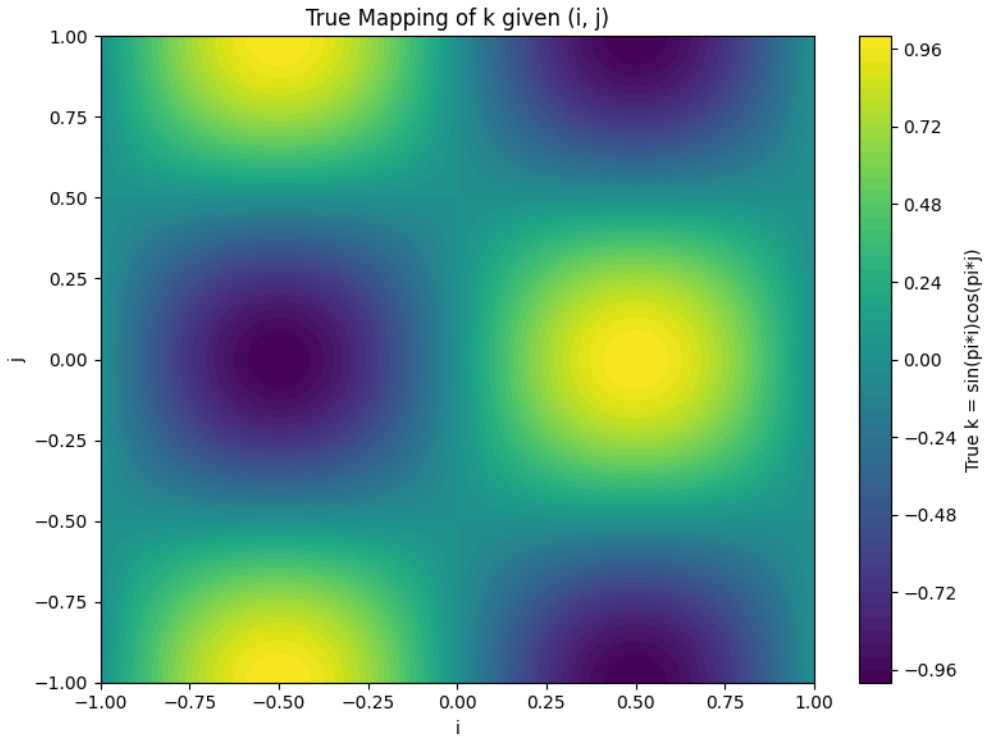

Figure 1: Learned mapping of k given (i, j) by the Flexible Conditional model.

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.