Neural Network Approaches for Learning Permutation Matrices in Column Subset Selection Problems

Harsh Kavediya Hanish Dhanwalkar
Vighnesh Nayak Somesh Ratre
Indian Institute of Technology, Bombay
{210100067,210100060,210100169,210260054}@iitb.ac.in

Abstract

This research proposes a novel approach to column subset selection problems by implementing continuous approximations of discrete steps in selection algorithms. We focus on the Linear Time Approximation Algorithm with Local Search, identifying discrete operations and developing differentiable alternatives to facilitate backpropagation in neural networks. Our work bridges the gap between combinatorial optimization techniques and gradient-based learning by developing continuous relaxations for key discrete operations including sampling, set membership updates, and matrix operations. We evaluate our approach on standard datasets and analyze the trade-offs between approximation quality and computational efficiency. This research contributes to the growing field of differentiable programming for combinatorial optimization problems and offers insights into learning permutation matrices with neural networks.

1. Introduction

Column Subset Selection Problems (CSSP) represent an important class of dimensionality reduction techniques that aim to select the most representative columns from a data matrix. Traditional approaches to CSSP rely on discrete algorithms that involve operations such as discrete sampling, set membership tests, and combinatorial optimization. While these methods provide strong theoretical guarantees, they present challenges for integration with modern deep learning frameworks due to their non-differentiable nature.

Recent advances in differentiable programming have opened new possibilities for solving combinatorial optimization problems within neural network architectures. By replacing discrete operations with continuous approximations, it becomes possible to leverage gradient-based optimization techniques while maintaining the structural properties of the original algorithms.

In this work, we focus on developing continuous approximations to the discrete steps in the Linear Time Approximation Algorithm for Column Subset Selection with Local Search. Our goal is to enable end-to-end training of neural networks for learning permutation matrices that represent optimal column subsets.

2. Problem Statement

2.1. Column Subset Selection Problem

Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ and an integer $k < n$, the Column Subset Selection Problem aims to find a subset of k columns from $\mathbf{A}$ that best represents the entire matrix. This can be formulated as finding a permutation matrix $\mathbf{P} \in \{0, 1\}^{n \times n}$ such that the first k columns of $\mathbf{AP}$ minimize the reconstruction error.

Formally, we seek to minimize:

$$\|\mathbf{A} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{A}\|_F^2 \tag{1}$$

where $\mathbf{C}$ represents the selected k columns of $\mathbf{A}$ and $\| \cdot \|_F$ denotes the Frobenius norm.

2.2. Challenges in Differentiable Approximation

The core challenge in developing a neural network approach to CSSP lies in the discrete nature of the permutation matrices. Key operations in CSSP algorithms include:

- Discrete sampling of column indices
- Binary set membership operations
- Discrete matrix updates based on selected columns

4. Proposed Approach

4.1. Algorithm Analysis

We begin by analyzing the Linear Time Approximation Algorithm for Column Subset Selection with Local Search, identifying the discrete steps that require continuous approximations:

4.1.1. Algorithm 1 (LSCSS): Discrete Steps

- **Sampling column indices:** Discrete selection of column index i with probability proportional to squared norm
- **Set membership update:** Adding element i to set I
- **Matrix update based on set I:** Creating zeros matrix D with specific diagonal entries
- **Set operations:** Emptying set I

4.1.2. Algorithm 2 (LS): Discrete Steps

- **Sampling column indices:** Discrete selection of $10k$ column indices
- **Uniform sampling:** Discrete selection of a single index
- **Set membership test:** Testing if indices exist in set I
- **Finding minimum index:** Discrete selection of minimizing index
- **Set operations:** Removing and adding elements to set I

4.2. Continuous Approximations

For each discrete step, we develop a differentiable approximation that preserves the essential properties while enabling gradient flow:

4.2.1. Sampling Approximations

- **Soft sampling:** Replace discrete sampling with a differentiable mechanism using the Gumbel-Softmax trick
- **Probability vector:** Create a probability vector based on column norms and apply temperature-controlled softmax
- **Continuous top-k selection:** Implement differentiable top-k via softmax normalization with temperature scaling
- **Soft uniform sampling:** Replace uniform sampling with a continuous relaxation using equal probabilities but soft selection

4.2.2. Set Membership Approximations

- **Soft membership:** Replace discrete set I with a continuous membership vector where each element has values between 0 and 1
- **Sigmoid functions:** Use sigmoid functions to approximate membership indicators
- **Continuous membership function:** Replace discrete membership test with a function measuring "degree of membership"

4.2.3. Matrix Operation Approximations

- **Weighted matrix operations:** Replace discrete selection matrices with soft selection matrices using membership weights
- **Continuous matrix approximation:** Matrix D can be approximated with a continuous function of the membership vector

4.2.4. Optimization Operation Approximations

- **Soft argmin:** Replace discrete argmin with differentiable soft-argmin function
- **Temperature scaling:** Use negative temperature-scaled softmax to approximate minimum selection
- **Continuous set updates:** Replace discrete set operations with continuous weights using element-wise operations on membership vectors

4.3. Neural Network Architecture

We propose a neural network architecture for learning permutation matrices:

- **Input processing:** A neural network that processes the input matrix $\mathbf{A}$ to produce score matrices
- **Differentiable sampling:** Implementation of the Gumbel-Softmax trick for sampling
- **Sinkhorn normalization:** Projection onto the Birkhoff polytope of doubly-stochastic matrices
- **Temperature annealing:** Gradual reduction of temperature parameter to sharpen distributions

5. Theoretical Analysis

5.1. Approximation Quality

We analyze the approximation quality of our continuous relaxations:

- **Sampling approximation:** As temperature $\tau \to 0$, the Gumbel-Softmax distribution converges to the categorical distribution
- **Set membership approximation:** Sigmoid functions with appropriate scaling approximate indicator functions
- **Matrix operation approximation:** Weighted operations converge to discrete operations as weights approach binary values

5.2. Convergence Properties

We establish theoretical guarantees on the convergence of our continuous approximations to the discrete algorithm:

- **Temperature annealing schedule:** Analysis of how different annealing schedules affect convergence
- **Gradient variance:** Characterization of gradient variance as a function of temperature
- **Approximation error bounds:** Establishing upper bounds on the error introduced by continuous approximations

6. Experimental Evaluation

6.1. Datasets

We evaluate our approach on standard datasets:

- **Synthetic datasets:** Matrices with controlled properties to analyze specific aspects of our approach
- **MNIST:** Handwritten digit dataset to evaluate performance on image data
- **Text corpora:** Document-term matrices to evaluate performance on text data

6.2. Evaluation Metrics

We use the following metrics to evaluate our approach:

- **Reconstruction error:** $\|\mathbf{A} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{A}\|_F^2$
- **Computational efficiency:** Running time and memory usage
- **Convergence behavior:** Analysis of convergence with different hyperparameters
- **Downstream task performance:** Evaluation on classification or regression tasks using the selected columns

6.3. Ablation Studies

We conduct ablation studies to analyze the contribution of each component:

- **Temperature sensitivity:** Impact of temperature parameter on approximation quality
- **Network architecture:** Comparison of different architectures for score prediction
- **Sampling mechanisms:** Comparison of different differentiable sampling approaches

7. Expected Results

We anticipate the following results from our experiments:

- **Approximation quality:** Our continuous approximations will achieve reconstruction errors within a small factor of the discrete algorithm
- **Computational efficiency:** The neural network approach will have higher training costs but faster inference times
- **Generalization:** The learned model will generalize to new matrices with similar statistical properties
- **Temperature effects:** Lower temperatures will yield more discrete solutions at the cost of higher gradient variance

8. Discussion and Future Work

Our work opens several avenues for future research:

- **Hybrid approaches:** Combining neural network initialization with discrete refinement
- **Alternative relaxations:** Exploring other relaxations such as SDP relaxations or message passing approaches
- **Application to other permutation problems:** Extending the approach to sorting, ranking, and matching problems
- **Theoretical foundations:** Developing tighter bounds on approximation quality and convergence rates

9. Conclusion

This proposal outlines a novel approach to column subset selection through continuous approximations of discrete algorithms. By developing differentiable alternatives to key operations in the Linear Time Approximation Algorithm, we enable end-to-end training of neural networks for learning permutation matrices. Our approach bridges the gap between combinatorial optimization and deep learning, offering new possibilities for solving permutation-related problems in a differentiable framework.

References

[1] Ahmed K Farahat, Ali Ghodsi, and Mohamed S Kamel. A fast linear time approximation algorithm for column subset selection with local search. *arXiv preprint arXiv:1303.0577*, 2013.

[2] Christos Boutsidis, Petros Drineas, and Michael W. Mahoney. Improved matrix column selection algorithms for nyström-based kernel learning. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 51–58, 2009.

[3] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *International Conference on Learning Representations*, 2018.

[4] Ryan P Adams and Richard S Zemel. Ranking using sinkhorn divergence. In *Advances in Neural Information Processing Systems*, pages 1368–1376, 2011.