Proximal Pairing and Scoring Algorithm

Blake Masters
Department of Computer Science
California Polytechnic State University
San Luis Obispo, United States of America
Email: bemaster@calpoly.edu

Borislav Hristov (Advisor)
Department of Computer Science
California Polytechnic State University
San Luis Obispo, United States of America
Email: bhristov@calpoly.edu

***Abstract*—This paper describes the Proximal Pairing algorithm for matching points between two matrices by aligning regional measurements. The algorithm iteratively establishes pairings using heuristic approaches such as selecting the closest item to each truth. Experimental results demonstrate improved accuracy when using Cosine Distance (1 - Sim) over Euclidean Distance. Pairings are evaluated through various scoring metrics, including identification label accuracy and percentage of points correctly paired.**

I. Introduction

The Proximal Pairing algorithm is used to match points between two matrices based on regional similarity measurements. The goal is to iteratively generate pairings using heuristic-based selection criteria. This approach has applications in data alignment, computer vision, and clustering tasks.

II. Algorithm Description

A. Notation

Let:

- M_1 and M_2 be two distance matrices of size $n \times n$.
- T be the set of known ground-truth pairings (i, j) where $i \in M_1$ and $j \in M_2$.
- P_k be the set of pairs generated through the algorithm.
- $D(i)$ be the distance between elements for index $i \in M_1$.

B. Proximal Pairing Algorithm

New pairs are iteratively added based on the following selection criteria:

1) **Closest:** Assigns a pair (i_1, j_1) by finding the indexes with the minimum distance in M_1 and M_2.
2) **Average:** Computes the average distance from all existing pairs and assigns the closest match:

$$\frac{\sum_{i \in M_1} D(i)}{n}, \quad \frac{\sum_{j \in M_2} D(j)}{n} \tag{1}$$

The algorithm proceeds as follows:

Algorithm 1 Proximal Pairing Algorithm

Input: Initial truth set T, distance matrices M_1, M_2
Initialize: Mark paired indices in M_1 and M_2
Set $P \leftarrow T$
while New pairs are found **do**
  Determine new pairs using selected heuristic:
  if Metric = Closest **then**
    Select (i_1, j_1) minimizing $D(i)$ and $D(j)$
  else if Metric = Average **then**
    Compute mean distance from all current pairs and assign closest
  end if
  Mark selected indices and update P
end while
Output: Set of new pairings P_k

III. Scoring Function

The generated pairings are evaluated based on:

- Proximity to ground-truth set.
- Distribution of distances between predicted pairs and ground truth.
- Computation of the final score:

$$S = \frac{1}{|P_k|} \sum_{(i,j) \in P_k} \text{rate}(i, j) \tag{2}$$

Algorithm 2 Scoring Function

Input: Set of predicted pairs P_k, ground truth T, distance matrices
Initialize: Scores list S
for each pair (i, j) in P_k **do**
  Compute the similarity rate based on distance metric
  Append score to S
end for
Compute final average score $S = \frac{1}{|P_k|} \sum S$
Output: Final score

IV. Experimental Setup

To evaluate the algorithm, experiments were conducted under three conditions:

A. Experiment 1: Correct Pairings

The initial k pairings are correct, providing an ideal benchmark.

B. Experiment 2: Incorrect Pairings

All initial k pairings are incorrect, testing robustness against erroneous initial conditions.

C. Experiment 3: Mixed Pairings

Half of the initial k pairings are correct, simulating real-world noisy datasets.

V. Experimental Results

Each experiment is iterated 30 times for robustness. The final performance is measured as:

$$\text{Score} = f(k) \tag{3}$$

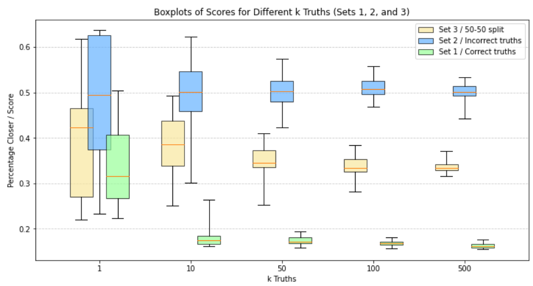

Fig. 1. Score variation with increasing k pairs

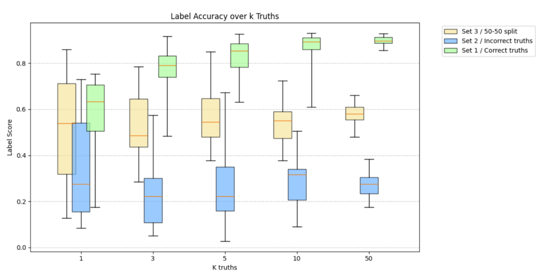

Fig. 2. Label Accuracy over k pairs

VI. Conclusion

The Proximal Pairing algorithm efficiently identifies optimal pairings in distance matrices. The scoring function evaluates different heuristics and selection strategies. Future improvements may include additional similarity metrics and adaptive pairing strategies.

Acknowledgment

This research was supported by the California Polytechnic State University Research Senior Project.

References