**Abstract**

Matching pairs of objects from different domains is a fundamental operation in data analysis. It typically requires the definition of a similarity measure between the classes of objects to be matched. For many cases, we may be able to design a cross domain similarity measure based on prior knowledge or to observe one based on the co-occurrence of such objects. In some cases, however, such a measure may not exist or it may not be given to us beforehand.

We develop an approach which is able to perform matching by requiring a similarity measure only within each of the classes. This is achieved by maximizing the dependency between matched pairs of observations by means of the Hilbert-Schmidt Independence Criterion. This problem can be cast as one of maximizing a quadratic assignment problem with special structure and we present a simple algorithm for finding a locally optimal solution.

**Problem Statement**

Assume we are given two collections of documents purportedly covering the same content, written in two different languages. Can we determine the document and its translation between these two sets of documents without using a dictionary?

(see Applications section on Multilingual Document Matching)

**Kernelized Sorting**

**Optimization problem**

\[ \pi^* = \arg \max_{\pi \in \Pi_m} \left[ K \pi^\top L \pi \right] \]

**Intuition**:

- \( K \) is a similarity measure (kernel matrix) on class of objects 1 and \( L \) is a similarity measure on class of objects 2. The more similar the two objects within class 1, the higher is the specific entry of \( K \), likewise for \( L \).
- The optimization above permutes objects in class 1 such that the dependency between objects in class 1 and class 2 is maximized.

**Facts**:

- The dependency measure being used is Hilbert-Schmidt Independence Criterion (Smola et. al., ALT, 2007).
- The optimization problem above belongs to quadratic assignment problem which is hard to solve.

**Sorting as a special case**

For scalar \( x_1 \) and \( y_1 \) and a linear kernel (similarity measure) on both classes, we can rewrite the optimization problem:

\[ \pi^* = \arg \max_{\pi \in \Pi_m} \left[ \pi \mathbf{1} \mathbf{1}^\top \pi \right] \]

The above means if objects in \( X \) are sorted in ascending order, objects in \( Y \) must be sorted in ascending order as well to maximize the dependency.

**Relaxation**

- To approximate the hard quadratic assignment problem, compute successive linear lower bounds and maximize:

\[ \pi_{t+1} = \arg \max_{\pi \in \Pi_m} \left[ K \pi_{t} \pi_{t}^\top L \pi_{t} \right] \]

This will converge to a local maximum.

- Initialization is done via sorted principal eigenvector.

**Image Matching**

Layout of 264 images into a 'ANU 2009' letter grid using Kernelized Sorting.

Class of objects 1 : image objects and class of objects 2 : Cartesian coordinate positions of the grid. One can see that images are laid out in the letter grid according to their color grading.

**Applications**

1: RSISE, ANU & SML, NICTA | 2: SCS, CMU | 3: Yahoo! Research

(Climbing hills and successive maximization analogy. The relaxation transforms the maximization problem into one with many small hills (many local maximum points). The small hill that we will end up at will depend on our starting point (initialization).)

**Multilingual Document Matching**

- Class of objects 1 : non-English documents (source languages) and class of objects 2 : its English translation (target language).
- Result comparisons with length based and dictionary based document matching.

**Data Attribute Matching**

- Class of objects 1 : first half of data attributes (or dimensions) and class of objects 2 : second half of data attributes.
- Results comparisons with random permutations and SVM classification/regression on original unsplitted data sets.

**Summary**

- We generalize sorting by maximizing dependency between matched pairs of observations via HSIC.
- Applications of our proposed sorting algorithm range from data visualization to image, data attribute and multilingual document matching.

1998) : Layout of 320 images into a 16 × 20 grid. GTM does not guarantee unique assignments of images to grids thus create blank spots.

Generative Topographic Mapping (Bishop et. al., Neural Computation, 1998) : Layout of 320 images into a 16 × 20 grid. GTM does not guarantee unique assignments of images to grids thus create blank spots.

Our Kernelized Sorting : Layout of 320 images into a 16 × 20 grid.

Novi Quadrianto | Le Song | Alex J. Smola