Learning Near-Isometric Matching Models

Julian J. McAuley* and Tibério S. Caetano

May 13, 2010

Abstract

Shape matching problems are typically described in terms of structural properties, encoding (for example) the ‘elasticity’ of certain joints in a rigid object. Unfortunately, encoding such constraints often results in a hard optimization problem (such as quadratic assignment), meaning that approximate solutions are typically employed; this presents a problem for many learning approaches that require exact inference schemes as a subroutine. We note that certain types of constraints, such as isometries, result in ‘easier’ optimization problems that can be solved with low tree-width graphical models. This allows us to apply learning in near-isometric matching scenarios, encoding rich first-order properties, as well as isometric and topological information.

Introduction

The well-known ‘quadratic assignment’ problem takes the form

$$\hat{f} = \arg\min_{f: A \rightarrow B} \sum_{a,b \in A} D_{a,b,f(a),f(b)},$$

where $f$ is an injective function. Many structural matching problems can be expressed in this form, for instance isometric matching can be written as

$$\hat{f} = \arg\min_{f: S \rightarrow T} \sum_{s_i, s_j \in S} |d(s_i, s_j) - d(f(s_i), f(s_j))|,$$

(1)

where $S$ and $T$ represent shapes (so that each $s_i$ defines the coordinates of a point), $f$ is an injective function mapping point coordinates in $S$ to point coordinates in $T$ (for instance, mapping the ‘joints’ in a deformable model to pixels in an image), and $d$ is the standard Euclidean distance function.

In cases where a zero-cost solution exists to (eq. 1), we note that far more efficient solutions to this problem can be found: we need not consider all edges between pairs of points in $S$, but rather a subset of edges that define a ‘globally rigid’ graph. We have investigated several graphical models whose embeddings in the plane are globally rigid. Examples are shown in Figure 1 (an ‘embedding’ is shown at right). Some examples demonstrating the matching problem are shown in Figure 2. These models continue to maintain good performance even as the assumption of a zero-cost solution becomes substantially violated.

Structured Matching Objectives

Given a graph $G$ with a globally rigid embedding, we can augment our potentials to encode structural constraints other than distance preservation. Generally, we write

$$\hat{f} = \arg\min_{f: S \rightarrow T} \sum_{i,j \in G} \langle \Phi_{i,j}(s_i, s_j, f(s_i), f(s_j)), \theta \rangle.$$

Here $\Phi$ is a feature vector describing the mapping from $(s_i, s_j)$ to $(f(s_i), f(s_j))$. One component of this vector would be $|d(s_i, s_j) - d(f(s_i), f(s_j))|$ as in (eq. 1), though other features could include topological information (for example in OCR applications where connectedness is important), and first-order properties such as Shape Contexts or SIFT features; we can also create higher-order features $\Phi_{i,j,k}$ encoding angle and scale information.

*The authors are with the Statistical Machine Learning Group at NICTA, and the Australian National University. Queries should be addressed to julian.mcauley@nicta.com.au.
model:

complexity: \( O(MN^4) \) \( O(MN^3) \) \( O(MN^2 \sqrt{N}) \) \( O(MN^2 \log N) \)

reference: [CC06] [MdCC10] [MCS08]

Figure 1: Some of our graphical models and their associated running times (for a template of size \(|S| = M\) in a scene of size \(|T| = N\); white nodes denote boolean variables. The nodes of this graph are ‘embedded’ in the plane, as shown for the model from [MCS08] at right. Our best methods are sub-cubic in \( N \), meaning that we improve upon the running time implied by the tree-widths of the graphs.

Figure 2: Some example matches obtained using various matching methods. Left: a first-order model using Shape Contexts; Center: a quadratic assignment model; Right: our model from [MCS08]. Incorrect assignments are shown in red.

\[ \theta \] is a parameter vector that controls the importance of the various features. We choose \( \theta \) using the structured learning framework of [THJA04]. Our objective function for choosing our parameter vector \( \hat{\theta} \) is given by

\[
\hat{\theta} = \arg\min_{\theta} \frac{1}{K} \sum_{i=1}^{K} \Delta(\hat{f}^i, f^i) + \frac{\lambda}{2}\|\theta\|^2_2 ,
\]

where \( f^1 \ldots f^K \) is our training set (consisting of labeled matches between our template and target shapes), \( \Delta(\hat{f}^i, f^i) \) is a loss function (encoding the error induced by choosing the assignment \( \hat{f}^i \) when the correct assignment is \( f^i \)), and \( \lambda \) is a regularization constant. This parameterization allows us to learn the varying degrees to which first-order properties as well as isometric and topological information are important in a specific matching scenario.

Our Findings

Our recent research has been focused on reducing the running time of exact inference beyond that implied by the tree-widths of the graphical models in question. Our sub-cubic algorithms are able to solve matching problems with thousands of nodes in a matter of seconds, by exploiting the specific structure of the potentials in isometric matching scenarios.

We find that models based on rigid graphs – in which inference can be done exactly – significantly outperform approximate methods based on quadratic assignment once learning is applied. This confirms the need for efficient, exact inference procedures in structured learning settings. Our model also outperforms first-order models based on Shape Contexts and SIFT features, confirming the need for high-order structural objectives in shape-matching problems.

References


