Graph Matching Framework

Ekaterina Tikhoncheva

This notes are a short description of graph matching model applied for finding feature correspondences between two images.

Contents

1 Problem statement

Consider two undirected weighted graphs $G^I = (V^I, E^I, A^I)$ and $G^J = (V^J, E^J, A^J)$, where V, E, A denote set of nodes, set of edges and set of node attributes respectively. We assume situation, where $|V^I| = n_1$, $|V^J| = n_2$ and n_1 is not necessary equal to n_2.
The aim of graph matching is to find a subset of possible node correspondences, which maximizes the similarity value between two graphs. Such subset can be represented by a binary vector $x \in \{0,1\}^{n_1 n_2}$, where $x_{(j-1)n_1+i} = 1$, if node $v_i \in V^I$ is matched to node $u_j \in V^J$, and $x_{(j-1)n_1+i} = 0$ otherwise. For simplicity we will write further x_{ij} instead of $x_{(j-1)n_1+i}$.
To measure similarity between graphs we define two similarity functions: *nodes similarity function* (first-order similarity) $s_V(v_i, u_j)$, $v_i \in V^I, u_j \in V^J$ and *edge similarity function* (second-order similarity) $s_E(e_{ii'}, e_{jj'})$, $e_{ii'} \in E^I, e_{jj'} \in E^J$. Both functions can be combined in one *similarity matrix* $S \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$, whose diagonal elements are $s_V(v_i, u_j)$ and non-diagonal elements are $s_E(e_{ii'}, e_{jj'})$.
Using this notation one can formulate *one-to-one Graph Matching Problem* **GMP** as an quadratic optimization problem ([1], [3], [2]):

$$\underset{x}{\operatorname{argmax}}\, x^T S x \tag{1}$$

$$\text{s.t. } x \in \{0,1\}^{n_1 n_2} \tag{2}$$

$$\sum_{i=1\ldots n_1} x_{ij} = 1 \tag{3}$$

$$\sum_{j=1\ldots n_2} x_{ij} = 1 \tag{4}$$

The maximum number of possible matches is equal to $\min(n_1, n_2)$. That means, in case when $n_1 \neq n_2$, only one of the conditions (3) or (4) will be fulfilled.
Quadratic Optimization Problem is known to be *NP*-hard [5]. This limits greatly the size of a graph, for which a exact solution can be calculated in reasonable time. Due to this there is a number of algorithms () that solve graph matching problem inexact. Unfortunately, most of the algorithms have two following problems:

1. they are still limited in size of permissible graphs. Experiments in most of the papers consider graphs with up to 100 nodes.

2. possible presence of outliers can reduce the accuracy of matching algorithm [6].

Our main aim was to develop a framework, which would allow an existing graph matching algorithm to cope with both problems.

2 Approach

The main idea of our approach is to create flexible multi-scale matching framework, that could improve the speed of theoretically any matching algorithm for large graphs and increase their accuracy.

Given an initial graphs G^I and G^J we create for each of them a coarse representative graphs A^I, A^J. Each node of such graph represent a subgraph of initial graph. We refer further to nodes of the coarse graphs A^I, A^J as *anchor nodes* or just *anchors* and to the graphs them self as *anchor graphs*.

Initial graphs with corresponding anchor graphs build two level structure (see fig ref): initial graphs (or *fine graphs*) represent a lower level and anchor graphs represent a higher level.

In case of large G^I and G^J graphs, where existing inexact solution techniques are difficult to apply due to time and storage complexity, their anchor graphs can be constructed small enough to not have this problems. That makes it possible, to find a solution of GMP on the higher level fast.

Ones the matching problem was solved on the higher level, we get the correspondences between subgraphs of the initial graphs, which are also much smaller comparing to complete graphs. Solving after that GMP for pairs of subgraphs lead us to the desirable correspondences between nodes of fine graphs.

Obviously, accuracy of such two level matching approach depends heavily on the partition of initial graphs into subgraphs and on the matching quality of anchor graphs. To make the approach robuster we integrate the result of matching on the lower level into graph partition algorithm and repeat the whole scheme several times till convergence.

In our work we concentrate ourself on the task of finding feature correspondences between two images. In the next sections we describe detailed the construction of initial graphs from given images and corresponding anchor graphs together with the matching algorithm on both levels and propagation of the matching results between the levels.

2.1 Lower Level Graph Construction

In this section we describe structure of a lower level graph built based on a given image. Extracted features of the image represent the nodes of the graph. The edges of the graph contain information about geometrical relations between the feature.

To extract features from the given image we detected first the edges ref of the image and used *dense SIFT descriptors* ref on the edge points. This results in hundreds or thousand of nodes.

2.2 Higher Level Graph Construction

2.3 Matching Algorithm

A standard approach to solve formulated problem approximately is to relax the integrality constrains: $x \in [0,1]^{n_1 n_2}$ instead of $x \in \{0,1\}^{n_1 n_2}$. To return back to discrete solution one can apply Greedy Matching or Hungarian Algorithm [4] on obtained continues solution.

2.4 Connection between two levels

References

[1] Minsu Cho and Olivier Duchenne. Finding Matches in a Haystack : A Max-Pooling Strategy for Graph Matching in the Presence of Outliers. *CVPR*, 2014.

[2] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted Random Walks for Graph Matching. *ECCV*, 2010.

[3] Minsu Cho and Kyoung Mu Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. *CVPR*, 2012.

[4] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterlyg*, 2:83–97, 1955.

[5] Sartaj Sahni. Computationally Related Problems. *SIAM J. Comput*, 3(4):262–279, 1974.

[6] Yumin Suh, Kamil Adamczewski, and Kyoung Mu Lee. Subgraph Matching using Compactness Prior for Robust Feature Correspondence. *CVPR*, 2015.