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* as an quadratic optimization problem ([1], [3], [2]):

$$\operatorname*{argmin}_x x^T S x \tag{1}$$

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

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

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

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 [**?**]. 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.
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 [**?**] on obtained continues solution.
Unfortunately, most of the algorithms are 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 ([**?**]).

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 perform graph matching on several stages. Given initial graphs G^I and G^J we create for each of them a coarse representative graph

2.1 Lower Level Graph Construction

2.2 Higher Level Graph Construction

2.3 Matching Algorithm

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. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 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. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.