Neurocomputing 134 (2014) 140–148

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A graph matching algorithm based on concavely regularized convex relaxation

Zhi-Yong Liu [a,*], Hong Qiao [a], Li-Hao Jia [a], Lei Xu [b]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China

ARTICLE INFO

Article history:
Received 29 May 2012
Received in revised form
25 October 2012
Accepted 31 December 2012
Available online 23 January 2014

Keywords:
Graph matching
Concave–convex procedure
Concave regularization
Frank–Wolfe algorithm
Convex relaxation

ABSTRACT

In this paper we propose a concavely regularized convex relaxation based graph matching algorithm. The graph matching problem is firstly formulated as a constrained convex quadratic program by relaxing the feasible set from the permutation matrices to doubly stochastic matrices. To gradually push the doubly stochastic matrix back to be a permutation one, an objective function is constructed by adding a simple weighted concave regularization to the convex relaxation. By gradually increasing the weight of the concave term, minimization of the objective function will gradually push the doubly stochastic matrix back to be a permutation one. A concave–convex procedure (CCCP) together with the Frank–Wolfe algorithm is adopted to minimize the objective function. The algorithm can be used on any types of graphs and exhibits a comparable performance as the PATH following algorithm, a state-of-the-art graph matching algorithm but applicable only on undirected graphs.

1. Introduction

Graph matching plays a central role in many graph based techniques. For instance, graph is frequently used as the structural representation of objects in computer vision and pattern recognition, and consequently the graph matching algorithm is commonly used to solve the object matching problem [5,1]. Graph matching involves identifying each vertex pair between graphs in some optimal way, or inherently finding a *good* permutation matrix between the two adjacency matrices of both graphs.[1] The problem is in nature a NP-hard combinatorial optimization problem with a factorial complexity, except for some graphs with special structure, such as the planar graphs, which has shown to be of polynomial complexity [13]. Therefore, an exhaustive search algorithm is computationally prohibited in practice, except for some small scale problems.

To make the problem computationally tractable, many approximate approaches have been proposed, trying to seek a good tradeoff between the complexity and matching accuracy. As summarized in [4], approximate matching algorithms can be roughly categorized into three groups, tree search based methods, spectral methods and continuous optimization (relaxation techniques). Tree search methods [22,6] are based on some simplifications of the depth-first search, for instance. Their performances depend largely on the problem nature, i.e., graph structure. The spectral methods [23,25] have their roots in the fact that the eigenvalues of the adjacency matrices of two isomorphic graphs are identical to each other. Unfortunately, the converse conclusion may be quite wrong, that is, two graphs with identical eigenvalues may be far from isomorphic. This might make the spectral methods result in a quite poor matching when the two graphs are not isomorphic.

Relaxation techniques involve relaxing the combinatorial matching problem to be a continuous one. The key point lies in the fact that optimization over a continuous set is usually easier to be approximated than its discrete counterpart. Specifically, the graph matching problem involves relaxing the set of permutation matrices, denoted by $\mathcal{P}$, to its convex hull, i.e., the set of doubly stochastic matrices denoted by $\mathcal{D}$. Typical relaxation techniques in the literature include for instance relaxation labeling [7,17], graduated assignment [9] and PATH following algorithm [28]. The relaxation labeling assigns each vertex of one graph with a probabilistic discrete label, and updates the label based on some measures, such as the vertex connectivity [7] or edit distance [17]. A common problem faced by the relaxation techniques is the backprojection which involves projecting the continuous solution found by the relaxed problem back to be a discrete one. Intuitively, given a $P_d \in \mathcal{D}$, the backprojection can be accomplished by a maximal linear assignment schema as given in (7), which is commonly employed by the relaxation labeling. However, such a linear projection may introduce a significant additional error. A soft assignment schema controlled by a parameter was introduced by

* Corresponding author.
E-mail address: zhiyong.liu@ia.ac.cn (Z.-Y. Liu).

[1] In this paper we consider only the equal-sized graph matching problem.

0925-2312/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.12.065

the graduated assignment algorithm to control the non-convexity of the problem [9]. As the parameter increases to be large enough, a permutation matrix is expected to be obtained though usually a clean-up step is further needed.

Different from the graduated assignment algorithm, the PATH following algorithm introduces a weighted linear combination of convex and concave relaxations to gradually get the discrete solution. Specifically, given the two graphs $G_D=(V_D,E_D)$ and $G_M=(V_M,E_M)$ to be matched where V and E, respectively, denote the sets of vertices and edges, it adopts the following square of Frobenius matrix norm as the objective function:

$$f(P)=\|A_D-PA_MP^T\|_F^2=\mathrm{tr}(A_D-PA_MP^T)^T(A_D-PA_MP^T), P\in\mathcal{P} \tag{1}$$

where A_D and A_M denote the adjacency matrices of G_D and G_M, respectively, $\mathcal{P}$ denotes the set of permutation matrix. By taking advantage of $P\in\mathcal{P}$, a convex relaxation of (1) can be found as follows [28]:

$$f_v(P)=\mathrm{vec}(P)^TQ\,\mathrm{vec}(P),\quad P\in\mathcal{D}, \tag{2}$$

where $\mathrm{vec}(P)$ creates a column vector from the matrix P by stacking the column vectors of P, and $Q=(I\otimes A_D-A_M^T\otimes I)^T(I\otimes A_D-A_M^T\otimes I)\in\mathbb{R}^{N^2\times N^2}$ is a symmetric definite positive matrix. The concave relaxation introduced by the PATH following algorithm is given by

$$f_c(p)=-\mathrm{tr}(\Delta P)-2\,\mathrm{vec}(P)^T(L_M^T\otimes L_D^T)\,\mathrm{vec}(P), P\in\mathcal{D}, \tag{3}$$

where $\Delta_{ij}=(D_M(i,i)-D_D(j,j))^2$, with D and L denoting the degree and Laplacian matrices of the graph, respectively. The concave relaxation holds the same minima as the original matching problem, but it is applicable only on undirected graphs without self-loops. Based on the convex and concave terms above, the objective function of the PATH following algorithm is given by

$$f_{path}(\gamma,P)=\gamma f_v(P)+(1-\gamma)f_c(P),\quad P\in\mathcal{D}, \tag{4}$$

where $\gamma\in[0,1]$ controls the non-convexity of the objective: a large γ means that $f_{path}(P,\gamma)$ tends to be convex; by contrast, a small γ makes $f_{path}(P,\gamma)$ concave. Thus, by gradually decreasing γ from 1 to 0, the objective becomes finally a concave one, and its minimization results in a permutation matrix. On equal-sized graph matching problems the PATH following algorithm exhibited a state-of-the-art performance in terms of both accuracy and complexity [28].

However, the PATH following algorithm cannot be used to solve the matching problem between directed graphs because the term in Eq. (3) can no longer guarantee to be concave. In this paper we introduce a much simpler concave term which can be applied on both directed and undirected graphs. Though the simple concave term is not a relaxation of the original matching problem, it is shown that it has a comparable performance as Eq. (3) on matching accuracy.[2] Moreover, instead of directly using the Frank–Wolfe algorithm, we firstly adopt the concave–convex procedure (CCCP) [27] to decompose the objective into a sequential constrained convex quadratic program, which is then solved by the Frank–Wolfe algorithm [8], avoiding the trouble of line search on a non-convex function. Section 2 is devoted to the proposed method, some experimental illustrations and discussions are given in Section 3, and finally Section 4 concludes the paper.

2. Proposed method

The objective function for the graph matching problem is firstly proposed, and then the CCCP together with Frank–Wolfe algorithm is proposed to minimize the objective, followed by an efficient initialization given by simplicial decomposition.

2.1. Objective function

The proposed objective function takes a similar form as Eq. (4), with the same convex relaxation but with a different concave term. To make the algorithm applicable for matching problems on both directed and undirected graphs, we propose to use the following concave term:

$$f_c(P)=-\mathrm{vec}(P)^T\,\mathrm{vec}(P),\quad P\in\mathcal{D}. \tag{5}$$

Then, similar to Eq. (4) an objective function of the graph matching problem is formulated as follows:

$$\min.\ f_\gamma(P)=\gamma\,\mathrm{vec}(P)^TQ\,\mathrm{vec}(P)-(1-\gamma)\,\mathrm{vec}(P)^T\,\mathrm{vec}(P),\quad \text{s.t. } P\in\mathcal{D}. \tag{6}$$

It is obvious that minimization of the concave term given by Eq. (5) results in an extreme point of $\mathcal{D}$, i.e., a permutation matrix. Thus, by gradually decreasing γ from 1 to 0, minimization of the objective will make P gradually converge to a permutation matrix.

At the beginning when $\gamma=1$, the objective function (6) degenerates to the convex relaxation, whose global minimization denoted by P_v can be obtained by the Frank–Wolfe algorithm (here we adopt the simplicial decomposition as discussed below). Actually, a permutation matrix can be directly obtained by an optimal linear assignment procedure which casts the doubly stochastic matrix P_v to be a permutation matrix P_p via

$$P_p=\arg\max_{P\in\mathcal{P}}\mathrm{tr}\,P_v^TP. \tag{7}$$

The assignment can be solved by the Hungarian algorithm [14], with a computational complexity $O(N^3)$. Such a *hard-cut* operation based graph matching algorithm, named QCV (quadratic convex), may however bring a big error into the final result, as to be witnessed by the experimental results in Section 3. By contrast, as γ gradually decreases, P is gradually pushed away from P_v in the way that update of P is guided to approach a permutation matrix with a smaller matching error. This point can be intuitively understood in the following way. During the convergence process the update direction of P comprises two parts, $g_v(P)$ and $g_c(P)$, the directions provided by the convex and concave terms, respectively. Guidance from $g_v(P)$ is to minimize the increase of the convex term, which, if can be globally minimized during the whole process, is equal to the difference between the best matching error and the global minimization of the convex relaxation got by P_v. On the other hand, $g_c(P)$ provides no informative search direction since any permutation matrix gives the same global minimum for the concave term. Thus, in the global minimization sense it is under the guidance of $g_c(P)$ that P is expected to approach a permutation matrix with a relatively small matching error.

To get an intuitive feel about the process, a simple example on matching two directed graphs with self-loops and with $N=3$ is given as shown in Fig. 1. P is parameterized as $P=[b,a,1-a-b;d,c,1-c-d;1-a-c,1-b-d,a+b+c+d-1]$ with the constraints $a\in[0,1],d\in[0,1],b\in[0,\pi],c\in[0,\pi]$ where $\pi=1-\max\{a,d\}$. In Fig. 1 the objective function is plotted by changing b and c with fixed a and d at their current estimations. As $\gamma=1$, P converges from the initial $1_{3\times3}/3$ to $a=0.462, d=0.305, b=0.374, c=0.451$, based on which the QCV gets the result as $P_{\in\mathcal{P}}=[0,1,0;1,0,0;0,0,1]$. As the algorithm proceeds, it is illustrated by the figure how P gradually approaches another solution with a smaller matching error, i.e., $P=I_3$, where $A_D=[0.496,0.302,0.826;0.179,0.390,0.876;0.037,0.998,0.999]$ and $A_M=[0.652,0.505,0.498;0.117,0.936,0.839;0.760,0.403,0.970]$.

[2] In our subsequent works we show theoretically that the simple concave term realizes exactly a concave relaxation.