

Efficient Graph Matching by Adjacency Matrix Based Model and Gaussian Smoothing Based Optimization

Xu Yang^{1,2}, Jing Yang^{3,1}, Yin-Lin Li¹, Zhang-Bin Zhou³, and Zhi-Yong Liu¹

¹State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

²Huizhou Advanced Manufacturing Technology Research Center Co., Ltd, Huizhou 516000, P. R. China

³School of Information Engineering, China University of Geosciences(Beijing), Beijing 100083, China

Abstract. Graph matching is a fundamental problem in information science and artificial intelligence, and can be applied to various data with structural characteristics. Approximate methods are necessary for most graph matching tasks due to their NP-complete nature. The continuous relaxation scheme offers an efficient approximate solution, whose performance can be improved by well tackling the nonconvexity of the objective function. Targeting at this problem, this paper proposes an efficient graph matching algorithm using adjacency matrix based graph matching model and generalizing Gaussian smoothing to approximately optimize it. Consequently, the proposed method can better reshape the objective function in the optimization process than existing algorithms, and meanwhile enjoy comparable efficient computation and storage. Experiments on both synthetic dataset and real-world image dataset validate the effectiveness of the proposed method.

Keywords: Graph matching, Gaussian smoothing, graduated optimization, adjacency matrix, graph algorithm

1 Introduction

Graph matching aims at finding an optimal set of vertex assignments between two graphs, which includes both the vertex difference and edge inconsistency. It is a fundamental problem in information science and artificial intelligence, and can be applied to various data with structural characteristics [4], such as computer vision data [2], chemical data [17], biological data [1], social network data [4], etc.

Graph matching is usually mathematically formulated by the quadratic assignment problem (QAP), which is known to be NP-complete [24, 25]. Therefore researchers mainly focus on efficiently finding an approximate solution by introducing certain relaxations.

In previous work [23], a path following (PF) algorithm is proposed by Zaslavskiy et al in 2009. It introduces the convex relaxation and concave relaxation function to objective function. And the continuous solution would automatically approach the discrete domain following a path by adjusting the parameter between the two functions. However, this method needs to construct the two complex functions explicitly. Based on PF algorithm, the graduated nonconvexity and concavity procedure (GNCCP) [10] is proposed in 2014. But GNCCP can realize the convex and concave procedure implicitly by adjusting a simple additional quadratic term and reshapes the objective function without considering its own form. The best choice for the convex (concave) relaxation function should be the convex (concave) envelope of original objective function, among infinite types of convex relaxations. But generating envelope would introduce a large amount of calculation. Gaussian smoothing is an optimal approximation for the envelope under certain conditions [14]. Yang et al [19] proposed a continuation method based on Gaussian smoothing which can be used to solve the regularized objective function of the combinatorial optimization problem in graph matching directly.

Inspired by the above observations, this paper targets at the adjacency matrix based graph matching model, and generalizes Gaussian smoothing to approximately optimize it. Consequently, the proposed method better reshapes the objective function in the optimization process without complex and specific deductions, and meanwhile enjoys comparable efficient computation and storage with the PF and GNCCP. The contributions are two-fold. First a regularized function incorporating the adjacency matrix based objective function and the combinatorial constraints is constructed for the application convenience of Gaussian smoothing, and second some closed-forms of functions in the Gaussian smoothing process are deduced.

The proposed method is loosely related to the PF and GNCCP, as they all use the adjacency matrix based model and graduated optimization. It is related to the Gaussian smoothing works from the optimization perspective [13–16]. Besides, our previous work also adopts Gaussian smoothing [20], which is applied to a different model.

The remaining manuscript is organized as follows: section 2 introduces the related works; The formulation of the adjacency matrix based model is first introduced in section 3, together with regularized function; The Gaussian smoothing based optimization process is introduced in section 4; After experimental evaluation in section 5, finally section 6 concludes the paper.

2 Related Works

Graph matching is known as an NP-hard problem which can be solved by approximate method. The continuous relaxation scheme forms an important group of approximate methods, which typically first relaxes the combinatorial constraints to be continuous, and then projects the continuous solution to the discrete domain. However, finding the optimum, or even the local optimum, of the relaxed continuous optimization problem is still a challenging problem, because of the high nonconvexity of the objective function.

Zaslavskiy et al proposed the path following (PF) algorithm [23] which used an adjacency matrix based QAP formulation, i.e. Koopmans-Beckmann's QAP, and achieves state-of-the-art performance and high computation and storage efficiencies. Specifically, it involves an $\mathcal{O}(N^3)$ computational complexity which is the same complexity with the multiplication operation between two adjacency matrices, and an $\mathcal{O}(N^2)$ storage complexity which is the same complexity with storing an adjacency matrix [12], where N is the vertex number. To tackle the nonconvexity, the PF algorithm construct a new objective function where the convex and concave function is combined by a parameter. And as the parameter decreases, the continuous solution would automatically approach the discrete domain following a path. The advantage of this graduated optimization procedure is the usually better discrete solution, because the projection to the discrete domain is supervised by the (relaxed) objective function. The disadvantage is the complex and specific deductions of the convex relaxation function and concave relaxation, which can hardly be generalized to other cases.

The graduated nonconvexity and concavity procedure (GNCCP) [10], proposed by Liu et al in 2014, avoids the complex and specific deductions by introducing a simple additional quadratic term. By adjusting the weight parameter of the quadratic term, the convex combination of the convex relaxation function and concave relaxation function is realized implicitly. This GNCCP actually provides a general combinatorial optimization framework, as its concise form makes it relatively easy to generalize to other combinatorial optimization problems, e.g. MAP on MRF [11]. It also can be applied to high order objective functions beyond the quadratic one. The cost of the simple form is that the additional term becomes independent from the objective function. Recently Mobahi and Fisher III proved that Gaussian smoothing can offer the optimal envelope of the nonconvex function approximately under certain conditions [14]. Based on the Gaussian smoothing, a continuation method is proposed to solve the objective function which is regularized by lagrange multiplier techniques directly.

3 Formulation Based on Adjacency Matrix

A labeled weighted graph $\mathcal{G} = \{V, E, L, W\}$ of size N consists of a vertex set $V = \{1, 2, \dots, N\}$, an edge set $E \subseteq V \times V$, a vertex label set $L = \{l_1, l_2, \dots, l_N\}$, and an edge weight set $W = \{w_1, w_2, \dots, w_P\}$, where P denotes the edge number. A vertex label $l_i \in \mathbb{R}^{|l| \times 1}$ may be a real number or real vector describing the attribute of the vertex i , where $|l|$ is its dimension. An edge weight $w_e \in \mathbb{R}^{|w| \times 1}$ describes the attribute of the edge $e = ij$, or say the link between vertices i and j , where $|w|$ is its dimension. The operation on a graph is generally realized by the exploration of the matrix associated with the graph. Weighted adjacency matrix is a common choice, which puts the edge weights of a graph into arrays. We take one-dimensional weight for instance, i.e. $w = 1$, which can be easily generalized to multi-dimensional case. The weighted adjacency matrix $\mathbf{G}$ associated with $\mathcal{G}$ is

$$\mathbf{G}_{ij} = \begin{cases} w_e, & \text{if an edge } e \text{ exist between vertices } i \text{ and } j, \\ \infty & \text{otherwise.} \end{cases} \quad (1)$$

In this paper we only consider undirected graph matching as in [7, 23], and therefore $\mathbf{G}$ is a symmetric matrix.

The matching between two graphs $\mathcal{G}$ and $\mathcal{H}$ means finding a set of assignments between the two vertex sets, which considers both vertex label difference and edge weight inconsistency. These assignments can be represented by an assignment matrix $\mathbf{X} \in \{0, 1\}^{N \times N}$, where $\mathbf{X}_{ia} = 1$ means assigning the vertex i in $\mathcal{G}$ to the vertex a in $\mathcal{H}$. Usually the one-to-one matching constraint is incorporated, which requires that a vertex in $\mathcal{G}$ can be at most assigned to at most a vertex in $\mathcal{H}$, and with this constraint $\mathbf{X}$ becomes a permutation matrix, that is

$$\mathbf{X} \in \Omega := \left\{ \mathbf{X} | \mathbf{X}_{ia} \in \{0, 1\}, \sum_i \mathbf{X}_{ia} = 1, \sum_a \mathbf{X}_{ia} = 1, \forall i, a. \right\} \quad (2)$$

The assignment matrix $\mathbf{X}$ can be obtained by the following combinatorial optimization problem:

$$\begin{aligned} \min D(\mathbf{X}) &= \alpha D_L(\mathbf{X}) + (1 - \alpha) D_W(\mathbf{X}) \\ \text{s.t. } \mathbf{X} &\in \Omega, \end{aligned} \quad (3)$$

where $D_L(\mathbf{X})$ and $D_W(\mathbf{X})$ respectively measure the vertex label difference and edge weight inconsistency, and weight parameter α is used to balance the two terms. The front term can be directly formulated by a linear term by

$$D_L(\mathbf{X}) = \text{tr}(\mathbf{L}^T \mathbf{X}), \quad (4)$$

where $\text{tr}(\cdot)$ denotes the matrix trace operation. The matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ stores the pre-calculated label difference measures between all the vertices in $\mathcal{G}$ and $\mathcal{H}$, and taking the Euclidean distance for instance there is

$$\mathbf{L}_{ia} = \|l_i^{\mathcal{G}} - l_a^{\mathcal{H}}\|_2, \quad (5)$$

where $\|\cdot\|_2$ means the L_2 norm. For the latter term $D_W(\mathbf{X})$, many graph matching algorithms choose to directly pre-calculate the difference measures between all edges in $\mathcal{G}$ and $\mathcal{H}$, and store them in a large affinity matrix [3, 5, 7, 8]. But such a matrix would lead to high computational complexity and storage complexity, especially when the input graphs are dense. The adjacency matrix based model would significantly reduce both types of complexity, and it typically takes the following form:

$$D_W(\mathbf{X}) = \|\mathbf{G} - \mathbf{X}\mathbf{H}\mathbf{X}^T\|_F^2, \quad (6)$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm.

Since $D_W(\mathbf{X})$ is a fourth order function with respect to $\mathbf{X}$, the PF algorithm transforms it to a quadratic function for deduction convenience of the convex relaxation function and concave relaxation function. Specifically, taking advantage of two permutation matrix properties, i.e. $\|\mathbf{A}\|_F = \|\mathbf{A}\mathbf{X}\|_F$ and $\mathbf{X}^T \mathbf{X} = \mathbf{I}$ with $\mathbf{I}$ denoting the identity matrix, $D_W(\mathbf{X})$ is transformed to

$$D'_W(\mathbf{X}) = \|\mathbf{G}\mathbf{X} - \mathbf{X}\mathbf{H}\|_F^2. \quad (7)$$

Though $D'_W(\mathbf{X})$ is equal to $D_W(\mathbf{X})$ when $\mathbf{X} \in \Omega$, the equality no longer holds when $\mathbf{X}$ is relaxed to be continuous, and it can hardly be figured out how much $D_W(\mathbf{X})$ is reshaped by this transformation, not to say the further convex relaxation and concave relaxation based on $D'_W(\mathbf{X})$. The GNCCP directly targets at the high order objective function $D_W(\mathbf{X})$ by implicitly combining the convex relaxation function and concave relaxation function through a simple additional term, which, however, is independently constructed without cues from $D_W(\mathbf{X})$. The Gaussian smoothing offers an alternative method for the high order term $D_W(\mathbf{X})$ with a better reshaping way in the optimization process.

3.1 Regularized Function

Since Gaussian smoothing is generally applied to the unconstrained continuous optimization problem, before applying Gaussian smoothing, a regularized function incorporating both the objective function and the combinatorial constraints is constructed as:

$$R(\mathbf{X}) = D(\mathbf{X}) + \zeta C(\mathbf{X}), \quad (8)$$

where ζ is the regularization parameter. The additional regularization term $C(\mathbf{X})$ is constructed according the constraints in Ω , which can be divided by

$$C(\mathbf{X}) = C_1(\mathbf{X}) + C_2(\mathbf{X}) + C_3(\mathbf{X}), \quad (9)$$

where

$$C_1(\mathbf{X}) = \sum_i \sum_a ((\mathbf{X}_{ia}^2)(\mathbf{X}_{ia} - 1)^2), \quad (10)$$

$$C_2(\mathbf{X}) = \sum_a (\sum_i \mathbf{X}_{ia} - 1)^2, \quad (11)$$

and

$$C_3(\mathbf{X}) = \sum_i (\sum_a \mathbf{X}_{ia} - 1)^2. \quad (12)$$

It can be easily verified that the minimal points of $C(\mathbf{X})$ in the unconstrained continuous domain locate and only locate in Ω , which implies that minimizing $C(\mathbf{X})$ would help push the continuous solution to Ω . Below we show how to minimize $R(\mathbf{X})$ by Gaussian smoothing.

4 Optimization by Gaussian Smoothing

The main idea of Gaussian smoothing is to form a sequence of subproblems from a simple one to the original optimization problem, and the continuous solutions of these subproblems would automatically approach a better local optimal point of the original optimization problem. The subproblems are formed by convolving $R(\mathbf{X})$ by an isotropic Gaussian kernel $G(\mathbf{X}, \sigma)$

$$\mathcal{R}(\mathbf{X}) = R(\mathbf{X}) * G(\mathbf{X}, \sigma), \quad (13)$$

where the kernel width σ is the same for all the dimensions and one-dimensional Gaussian kernel is

$$g(\mathbf{X}_{ia}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{X}_{ia}\|^2}{2\sigma^2}\right). \quad (14)$$

If σ is large enough, the nonconvex parts of $R(\mathbf{X})$ would be smoothed, and the initial simple subproblem becomes a convex optimization problem. Then by gradually decreasing σ to 0, minimizing $\mathcal{R}(\mathbf{X})$ becomes equivalent to minimizing $R(\mathbf{X})$.

Generating the convolved function for an arbitrary nonconvex function would be computationally expensive. Thanks to the polynomial form of $R(\mathbf{X})$, there exists closed-form for $\mathcal{R}(\mathbf{X})$. First $\mathcal{R}(\mathbf{X})$ can be divided into

$$\mathcal{R}(\mathbf{X}) = \mathcal{D}(\mathbf{X}) + \zeta \mathcal{C}(\mathbf{X}), \quad (15)$$

which two terms are respectively the convolved functions of $D(\mathbf{X})$ and $C(\mathbf{X})$. To get the closed-form of the convolved function $\mathcal{D}(\mathbf{X})$, the high order term $D_W(\mathbf{X})$ in $D(\mathbf{X})$ needs first to be expanded, which begins with

$$D_W(\mathbf{X}) = \|\mathbf{G} - \mathbf{X}\mathbf{H}\mathbf{X}^T\|_F^2 = \sum_i \sum_j (\mathbf{G}_{ij} - \sum_a \sum_b \mathbf{X}_{ia} \mathbf{H}_{ab} \mathbf{X}_{jb}). \quad (16)$$

Then it is further expanded to the sum of all polynomial terms, and the Gaussian smoothing operation is applied to each of these terms. Omitting the deduction details, the final convolved function $\mathcal{D}_W(\mathbf{X})$ after reorganization to the matrix form is

$$\mathcal{D}_W(\mathbf{X}) = \|\mathbf{G} - \mathbf{X}\mathbf{H}\mathbf{X}^T\|_F^2 + N\sigma^2 \text{tr}(\mathbf{X}(\mathbf{H}\mathbf{H}^T + \mathbf{H}^T\mathbf{H})\mathbf{X}^T) + \sigma^4 \|\mathbf{H}\|_F^2. \quad (17)$$

If the reader would like to make more investigations in the above deduction, one should note that the fourth order term with respect to a single variable $\mathbf{X}_{ia}$ is not considered in the deduction, where we make use of the characteristics of $\mathbf{G}$ and $\mathbf{H}$ that their diagonal terms are all 0. To be more specific, the only case there exists a fourth order term with respect to the single variable $\mathbf{X}_{ia}$ is when $i = j$ and $a = b$, and meanwhile the corresponding polynomial term is always 0 by substituting $\mathbf{G}_{ij} = 0$ and $\mathbf{H}_{ab} = 0$ into (17). Since the third term $\sigma^4 \|\mathbf{H}\|_F^2$ in $\mathcal{D}_W(\mathbf{X})$ is a constant with respect to $\mathbf{X}$, it can be ignored in the later optimization process. The convolved function $\mathcal{D}_L(\mathbf{X})$ of the linear term is exactly $D_L(\mathbf{X})$ itself. After removing irrelevant constant there is

$$\begin{aligned} \mathcal{D}(\mathbf{X}) &= D(\mathbf{X}) + N\sigma^2 \text{tr}(\mathbf{X}(\mathbf{H}\mathbf{H}^T + \mathbf{H}^T\mathbf{H})\mathbf{X}^T) \\ &= D(\mathbf{X}) + 2N\sigma^2 \text{tr}(\mathbf{X}(\mathbf{H}\mathbf{H}^T)\mathbf{X}^T). \end{aligned} \quad (18)$$

The second equation holds because of the undirected assumption mentioned above. Similar to [20], in the deduction of $\mathcal{C}(\mathbf{X})$, after the constant removal there is

$$\mathcal{C}_1(\mathbf{X}) = C_1(\mathbf{X}) + 6\sigma^2 \sum_i \sum_a (\mathbf{X}_{ia}(\mathbf{X}_{ia} - 1)), \quad (19)$$

and

$$\mathcal{C}_2(\mathbf{X}) = C_2(\mathbf{X}), \quad \mathcal{C}_3(\mathbf{X}) = C_3(\mathbf{X}). \quad (20)$$

To sum up, finally there is

$$\mathcal{R}(\mathbf{X}) = R(\mathbf{X}) + 2N\sigma^2 \text{tr}(\mathbf{X}(\mathbf{H}\mathbf{H}^T)\mathbf{X}^T) + 6\zeta\sigma^2 \sum_i \sum_a (\mathbf{X}_{ia}(\mathbf{X}_{ia} - 1)). \quad (21)$$

It can be proved that in (21) the latter two additional terms are convex with respect to $\mathbf{X}$. Specifically, in the second term, there is

$$\text{tr}(\mathbf{X}(\mathbf{H}\mathbf{H}^T)\mathbf{X}^T) = \text{vec}(\mathbf{X})^T (\mathbf{I}_{NN} \otimes (\mathbf{H}\mathbf{H}^T)) \text{vec}(\mathbf{X}), \quad (22)$$

where $\text{vec}(\mathbf{X})$ is the row-wise replica of $\mathbf{X}$, the matrix $\mathbf{I}_{NN} \in \{0, 1\}^{N \times N}$ is an identity matrix, and the operation $\otimes$ is the Kronecker product between two matrices. Since $\mathbf{H}$ is a symmetric matrix, $\mathbf{H}\mathbf{H}^T$ is a positive semi-definite matrix. Therefore $\mathbf{I}_{NN} \otimes (\mathbf{H}\mathbf{H}^T)$ is positive semi-definite, as the Kronecker product between symmetric positive semi-definite matrices is also positive semi-definite [6]. Given that N and σ are nonnegative, the second term is thus a convex function. The third term is also convex as each polynomial term $\mathbf{X}_{ia}(\mathbf{X}_{ia} - 1)$ in the sum is convex.

The sequence of subproblems in Gaussian smoothing process can be obtained by decreasing σ from ∞ to 0. When $\sigma \rightarrow \infty$, minimizing $\mathcal{R}(\mathbf{X})$ is equivalent to the following minimization problem:

$$\min \mathcal{R}_\infty(\mathbf{X}) = 2N\sigma^2 \text{tr}(\mathbf{X}(\mathbf{H}\mathbf{H}^T)\mathbf{X}^T) + 6\zeta\sigma^2 \sum_i \sum_a (\mathbf{X}_{ia}(\mathbf{X}_{ia} - 1)), \quad (23)$$

which is a convex optimization problem based on the above analysis. Therefore its global solution exist, which can be efficiently found by the gradient descent method. In implementation, by initializing $\mathbf{X}$ by this global solution, it only needs to initialize σ to be a large positive real rather than ∞ . And $\mathbf{X}$ for each of the subsequent subproblems is initialized by the previous local solution, which is also minimized by the gradient descent method. The gradient for $\mathcal{R}(\mathbf{X})$ is

$$\nabla \mathcal{R}(\mathbf{X}) = \nabla D(\mathbf{X}) + \zeta \nabla C(\mathbf{X}) + 4N\sigma^2 \mathbf{X}\mathbf{H}\mathbf{H}^T + 6\zeta\sigma^2(2\mathbf{X} - \mathbf{1}_{NN}), \quad (24)$$

where $\mathbf{1}_{NN} \in \{0, 1\}^{N \times N}$ is a matrix with all 1 entries. For $\nabla D(\mathbf{X})$ and $\nabla C(\mathbf{X})$ there are

$$\nabla D(\mathbf{X}) = \alpha \mathbf{L} + (1 - \alpha)(2\mathbf{X}\mathbf{H}(\mathbf{X}^T \mathbf{X} \mathbf{H}^T + \mathbf{X} \mathbf{H}^T \mathbf{X}^T) - 2(\mathbf{G}\mathbf{X}\mathbf{H}^T + \mathbf{X}\mathbf{H}\mathbf{G}^T)), \quad (25)$$

and

$$\nabla C(\mathbf{X}) = 2\mathbf{X} \circ (\mathbf{X} - \mathbf{1}_{NN}) \circ (2\mathbf{X} - \mathbf{1}_{NN}) + 2(\mathbf{X}\mathbf{1}_{NN} + \mathbf{1}_{NN}\mathbf{X}) - 4\mathbf{1}_{NN}. \quad (26)$$

Note in (25) the multiplication $\mathbf{X}^T \mathbf{X}$ is no longer equal to I when it is relaxed to the continuous domain.

The whole optimization process is summarized in Algorithm 1. The computational complexity of the algorithm is the same with PF and GNCCP, which is $\mathcal{O}(N^3)$, while the storage complexity is as low as $\mathcal{O}(N^2)$.

Algorithm 1 Gaussian smoothing based optimization

Input: Two graphs $\mathcal{G}$ and $\mathcal{H}$;

Output: An assignment matrix $\mathbf{X} \in \Omega$;

- 1: Construct adjacency matrices $\mathbf{G}$ and $\mathbf{H}$;
 - 2: Initialize $\mathbf{X}_0$ by solving the convex optimization problem (23);
 - 3: Initialize σ ;
 - 4: **while** $\sigma \geq 0$ **do**
 - 5: Solving $\mathbf{X}^* = \arg \min_{\mathbf{X}} \mathcal{R}(\mathbf{X})$ by the gradient descent method with the initialization of $\mathbf{X}_0$;
 - 6: $\mathbf{X}_0 \leftarrow \mathbf{X}^*$;
 - 7: $\sigma = \sigma - \delta$ where δ is small positive value;
 - 8: **end while**
-

5 Experiments

The proposed method is evaluated on both synthetic dataset and real-world image dataset, by comparing it with two other adjacency matrix based methods, i.e. the classic Umeyama's algorithm (Ume) [18] and the state-of-the-art PF algorithm.

5.1 Synthetic dataset

In this experiment, synthetic points are generated by random sampling with respect to different changing factors, i.e. noise level, problem scale, and edge density, which are then represented by graphs. The first point set $G = \{g_1, g_2, \dots, g_N\}$ is generated by sampling each point $g_i \in [0, 1]^{1 \times 2}$ from a two dimensional uniform distribution; Then a ground truth assignment matrix $\mathbf{X}^{gt}$ is randomly sampled from the set of permutation matrices; The second point set $H = \{h_1, h_2, \dots, h_N\}$ is obtained based on G and $\mathbf{X}^{gt}$ by the following transformation

$$H_a = g_i + \eta, \text{ if } \mathbf{X}_{ia}^{gt} = 1, \quad (27)$$

where the additional noise $\eta \in \mathcal{N}(0, \sigma_\eta)^{1 \times 2}$ is sampled from the two-dimensional isotropic Gaussian distribution. Each point is represented by a graph vertex, the vertex label is not considered in this experiment for a pure comparison of the graph structure matching performance. The graph structure is built in a sparse manner following our pervious works [20,21], i.e. by randomly adding and removing $\sigma_\eta P$ edges in each graph with the edge number denoted by P .

Three simulations are performed with respect to the three changing factors including noise level, problem scale, and edge density, with the other two fixed when changing one factor. For the *noise level* simulation, σ_η is increased from 0 to 0.1 by a step size of 0.01; For the *problem size* simulation, N is increased from 20 to 40 by a step size of 2; For the *edge density* simulation, the edge density ρ is increased from 0.1 to 1 by a step size 0.1. The comparison results are shown in Figure 1. It can be observed that in the front two simulations, the proposed method achieves comparable or slightly better performance than the state-of-the-art PF algorithm. In the third simulation, when the edge density is low, e.g. $\rho = 0.1 \sim 0.2$, neither PF nor OUR achieve perfect matching, which implies that the too sparse connections between vertices could not offer enough structural constraints. Also note that when the vertices are completely connected, i.e. $\rho = 1$, the matching performances deteriorate for PF and OUR, which may be because the too many connections further introduce additional error. In the three simulations, Ume achieves a good performance only when the $\sigma_\eta = 0$, which means that it is inapplicable to the noisy situation.

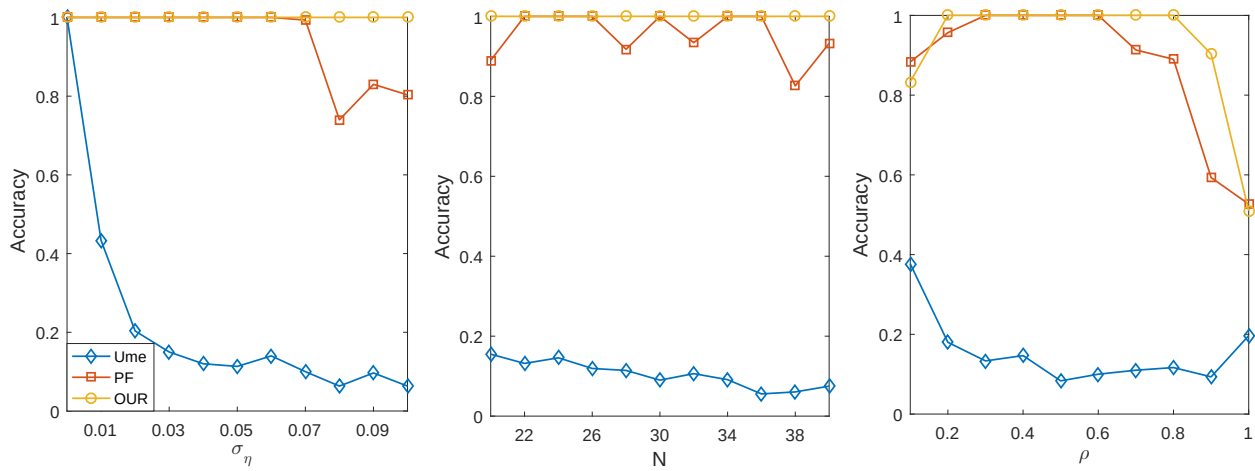


Fig. 1. Results on synthetic dataset. The left, middle, and right subfigures respectively correspond to the simulations w.r.t. the noise level, problem scale, and edge density.

5.2 Handwritten Chinese character matching dataset

The proposed method is also applied to a handwritten Chinese character matching dataset. The dataset consists of 40 images associated with 4 Chinese characters fetched from [9], with 10 images for each character, as illustrated in Figure 2. The experimental settings generally follow [22]. Specifically, images associated with the first and third characters are manually labeled with 28 ground truth points, and those associated with the second and fourth characters are labeled with 23 ground truth points. The shape context feature, a simple descriptor calculating the point distribution around each point, is used as the vertex label, and the graph structure is built according to the character skeleton.

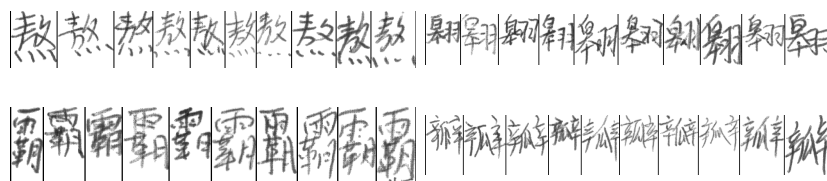


Fig. 2. Samples from handwritten Chinese character matching dataset.

The results are shown in Table 1, which validate the state-of-the-art performance of the proposed method. For the first character PF outperforms OUR, while for the latter three characters OUR achieves comparable performance with PF. The advantage of the proposed method is that it can be easily generalized to other cases without specific deduction of the convex and concave relaxation functions. Some matching samples are presented in Figure 3.

Table 1. Average matching accuracy (%) on images associated with different handwritten Chinese characters.

Character\Algorithm	Ume	PF	OUR
Character #1	15.0	100.0	98.6
Character #2	23.5	100.0	100.0
Character #3	21.2	100.0	100.0
Character #4	37.4	98.3	98.3

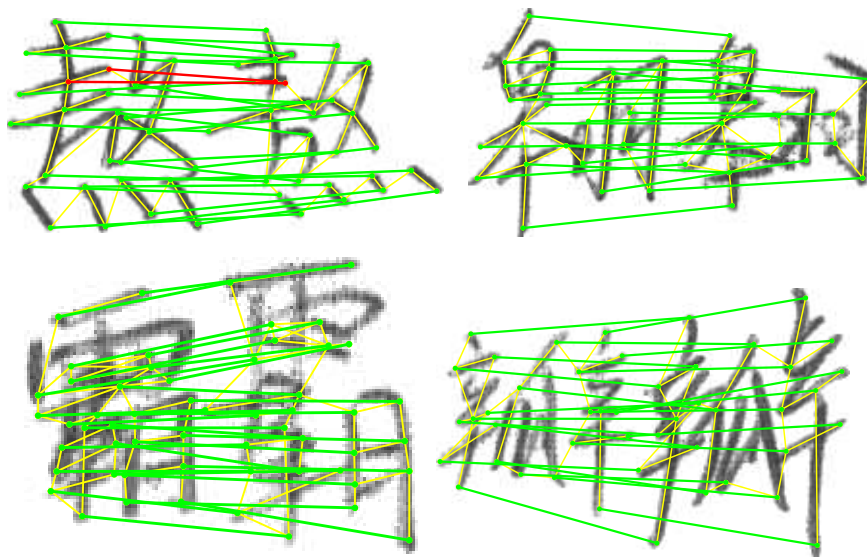


Fig. 3. Matching samples on handwritten Chinese character matching dataset. The vertices are in green, and the graph structures are denoted by yellow lines. The green lines denote correct assignments, while the red ones denote incorrect assignments.

6 Conclusion

This paper proposes a graph matching algorithm, which could take advantage of the efficiency of the adjacency matrix based model and meanwhile well tackle its high nonconvexity. The contributions mainly lie in the construction of the regularized function and the deduction of the closed-forms of the functions in the Gaussian smoothing process. Finally, the effectiveness of the proposed method is validated by experiments on synthetic dataset and real-world dataset.

References

1. M.S. Amin, R.L. Finley, and H.M. Jamil. Top-k similar graph matching using tram in biological networks. *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 9(6):1790–1804, 2012.
2. A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, pages 26–33, 2005.
3. M. Cho, J. Lee, and K.M. Lee. Reweighted random walks for graph matching. In *Proc. European Conf. Computer Vision*, pages 492–505, 2010.
4. D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *Int'l J. pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

5. T. Cour, P. Srinivasan, and J.B. Shi. Balanced graph matching. In *Proc. Advances in Neural Information Processing Systems*, pages 313–320, 2006.
6. G.H. Golub and C.F.V. Loan. *matrix computations*. Johns Hopkins University Press, 1996.
7. M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 1482–1489, 2005.
8. M. Leordeanu, R. Sukthankar, and M. Hebert. Unsupervised learning for graph matching. *Int'l J. Computer Vision*, 96(1):28–45, 2012.
9. C.L. Liu, F. Yin, D.H. Wang, and Q.F. Wang. CASIA online and offline Chinese handwriting databases. In *Proc. Int'l Conf. Document Analysis and Recognition*, pages 37–41, 2011.
10. Z.Y. Liu and H. Qiao. GNCCP - graduated nonconvexity and concavity procedure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36(6):1258–1267, 2014.
11. Z.Y. Liu, H. Qiao, and J.H. Su. Map inference with mrf by graduated non-convexity and concavity procedure. In *Int'l Conf. Neural Information Processing*, pages 404–412, 2014.
12. Z.Y. Liu, H. Qiao, and L. Xu. An extended path following algorithm for graph-matching problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(7):1451–1456, 2012.
13. M. Loog, J. JisseDuistermaat, and L.M.J. Florack. On the behavior of spatial critical points under gaussian blurring a folklore theorem and scale-space constraints. In *Scale-Space and Morphology in Computer Vision*, pages 183–192, 2001.
14. H. Mobahi and J.W. Fisher. On the link between gaussian homotopy continuation and convex envelopes. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 43–56, 2015.
15. M. Nielsen. Graduated non-convexity by smoothness focusing. In *Proc. British Machine Vision Conference*, pages 60.1–60.10, 1993.
16. A. Shameli and Y. Abbasi-Yadkori. A continuation method for discrete optimization and its application to nearest neighbor classification. *arXiv:1802.03482*, available at <https://arxiv.org/abs/1802.03482>.
17. A. Smalter, J. Huan, and G. Lushington. GPM: A graph pattern matching kernel with diffusion for chemical compound classification. In *Proc. IEEE Int'l Conf. BioInformatics and BioEngineering*, pages 1–6, 2008.
18. S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.
19. X. Yang, Z. Y. Liu, and H. Qiao. A continuation method for graph matching based feature correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Accepted and to Appear.
20. X. Yang, Z.Y. Liu, and H. Qiao. A continuous method for graph matching based on continuation. In *Int'l Conf. Intelligence Science and Big Data Engineering*, 2018, accepted and to appear.
21. X. Yang, H. Qiao, and Z.Y. Liu. Feature correspondence based on directed structural model matching. *Image and Vision Computing*, 33:57–67, 2015.
22. X. Yang, H. Qiao, and Z.Y. Liu. Outlier robust point correspondence based on GNCCP. *Pattern Recognition Letters*, 55:8–14, 2015.
23. M. Zaslavskiy, F. Bach, and J.P. Vert. A path following algorithm for the graph matching problem. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2009.
24. F. Zhou and F. De la Torre. Factorized graph matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 127–134, 2012.
25. F. Zhou and F. De la Torre. Deformable graph matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2922–2929, 2013.