A comparison of Junction Tree and Relaxation Algorithms for point matching using different distance metrics

Tibério S. Caetano[†‡],Terry Caelli[†] and Dante A. C. Barone[‡]
{tcaetano,tcaelli}@cs.ualberta.ca, barone@inf.ufrgs.br

†Department of Computing Science, University of Alberta, Edmonton, AB, Canada, T6G 2E8
‡Instituto de Informática, UFRGS, Porto Alegre, RS, Brazil, 91501-970

Abstract

We have developed a polynomial time optimal method for a class of attributed graph matching problems using the Junction Tree algorithm from Graphical Models. In this paper we compare this method with standard probabilistic relaxation labeling using different forms of point metrics and under different levels of additive noise. Results show that, no matter which of the metrics is applied, our technique is more effective than probabilistic relaxation labeling for large graph sizes. For small graph sizes, our technique is still preferable for two of the metrics, while for the third one both techniques perform similarly.

1. Introduction

The abstract and general nature of the attributed graph matching problem makes it a powerful tool for modeling a variety of pattern matching tasks where complex structures are involved. A particular and important application domain is point pattern matching. It occurs in several domain instances, most noticeably in stereo matching [7]. In point matching, the edge attributes are considered to be functions of the Euclidean distance between the points [2], while vertex attributes are typically ignored when some type of invariant matching is required.

In a very recent study, we have explored a hidden Markov Random Field formulation and the Junction Tree algorithm to find a global optimum for the point matching problem in polynomial time [1]. In that work, we modeled the edge attributes as a Gaussian function of the relative Euclidean distances between the points and used triangulated distances between the vertex sets. This model proved to be quite robust to noise and had a perfect accuracy which was insensitive to the size of the involved graphs when in the absence of noise.

The contribution of the present paper is to apply to this technique some additional types of proximity measures between the points (as well as the Gaussian itself), and to compare the performance of this model to the well-known technique of probabilistic relaxation labeling [3]. Results obtained clearly showed that, for large graphs, our technique outperforms probabilistic relaxation labeling for all the alternative proximity measures explored. For small graphs, our technique still performs better when two of the three proximity measures are used. The two methods perform similarly for small graphs using the third measure.

2. The Problem

We consider the problem of matching two point patterns in $\mathbb{R}^2$ invariant to translations, rotations and reflections. In this work, the scale of both point patterns must be the same. This is the problem of *isometry-invariant* point matching, or matching under transformations that preserve length [1].

The only information that is available is the *position* of each point with respect to some reference frame. For clarity, we call the pattern which consists in the domain of the mapping as “domain pattern” and the other as “codomain pattern”. There are several ways of formally defining this problem, depending on the type of constraints enforced in the mapping. In this paper, the only constraint enforced is that the function (mapping) must be “total”: every element in the domain pattern must map to some element in the codomain pattern (but the opposite may not be true), since we have shown in [1] that, using this constraint, it is possible to achieve optimal matching in polynomial time with our method.

3. Review of Optimal Point Matching

In this section we provide a brief overview of the optimal point matching approach presented in [1].

The points in the domain and in the codomain are associated to nodes of graphs G_t and G_s, respectively. The relative distance between a pair $\{i, j\}$ of points in a pattern is seen as an *edge attribute* of the edge that connects vertices i and j in G_t or G_s. In this formulation, point pattern matching turns out to be an attributed graph matching problem.

It is possible to show that encoding only those distances corresponding to the edges of a triangulated (formally a “3-tree”) mesh over a point set is sufficient to uniquely encode the complete distance matrix. This property, in conjunction with the known optimality of the Junction Tree algorithm for inference in triangulated graphical models, allowed us to create this new optimal matching algorithm that runs in polynomial time [1].

The model formulation consists, initially, in defining each of the T vertices in G_t as a random variable (*rv*) that can assume S possible values, corresponding to the vertices in G_s. In this way, we can define the compatibility of a joint match of two *rv*’s x_i and x_j in G_t to values θ_k and θ_l in G_s as

$$\psi_{ij;kl} = P(x_i = \theta_k | x_j = \theta_l)$$

or, in matrix form, for each pair $\{i, j\}$ in G_t, we obtain a *proximity matrix*,

$$\psi_{ij} = p(x_i|x_j) = \frac{1}{C}\begin{pmatrix} \mathcal{P}(y_t^{ij}, y_s^{11}) & \dots & \mathcal{P}(y_t^{ij}, y_s^{1S}) \\ \vdots & \ddots & \vdots \\ \mathcal{P}(y_t^{ij}, y_s^{S1}) & \dots & \mathcal{P}(y_t^{ij}, y_s^{SS}) \end{pmatrix}$$

where y_a^{bc} denotes the binary attribute between vertices b and c in graph G_a (which is the distance from the corresponding points in the point pattern). The key point, however, is that the edge connections in the model are drawn as a 3-tree, and not between every pair of nodes. This retains the entire information and, at the same time, generates a simple model graph whose maximal clique size is 4, what has implications for the time complexity of the matching algorithm, as we will see. C is a normalization constant that equals the sum of all elements in the matrix, in order to keep ψ_{ij} compatible with a probability distribution. The operator $\mathcal{P}$ is a proximity function that measures the similarity of the two arguments. Several options are available for $\mathcal{P}$ [2], and in [1] we have explored the univariate Gaussian function:

$$\mathcal{G}(y_t^{ij}, y_s^{kl}) = \exp\left(-\frac{1}{2\sigma^2}||y_t^{ij} - y_s^{kl}||^2\right).$$

The resulting model is shown in Figure 1. Optimal matching is then obtained by running an exact inference engine in this graphical model. We have used the HUGIN algorithm [6], an instance of the Junction Tree (JT) framework [5]. This algorithm enables optimal inference over triangulated graphs and propagates optimal evidence for node states (in our case, vertex labels of one graph in the other as a function of the compatibilities between vertices and edges). Evidence is propagated via marginalization defined over separator nodes that define relations between cliques. The complexity of the resulting optimal algorithm turns out to be exponential on the size of the maximal clique of the model. In our case, it is $\mathcal{O}(S^4T)$, thus polynomial in the size of both graphs. Key to the model’s efficiency is the using of lower-order cliques instead of the complete distance matrix, as is the case in techniques like probabilistic relaxation labeling.

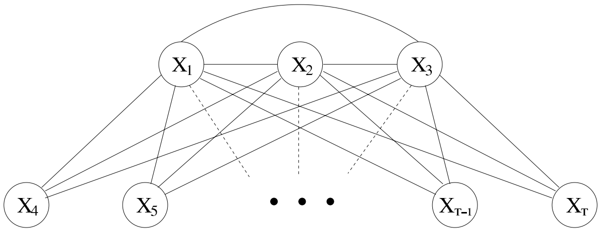


Figure 1. A possible 3-tree model.

4. Alternative Proximity Functions

In a recent work, Carcassoni and Hancock presented several types of point proximity functions for point matching [2]. Here we consider some similar alternative choices in our evaluation of the robustness of our approach to different types of proximity functions. We use a notation similar to that of [2].

A possible choice for a proximity function is based on the Hyperbolic Tangent function, and can be written as

$$\mathcal{H}(y_t^{ij}, y_t^{kl}) = 1 - \tanh\left[\frac{||y_t^{ij} - y_s^{kl}||}{\sigma}\right].$$

The second alternative is an Increasing Weighting function, which is simply

$$\mathcal{I}(y_t^{ij}, y_s^{kl}) = \left[1 + \frac{||y_t^{ij} - y_s^{kl}||}{\sigma}\right]^{-1}.$$

Each of these types of point proximity functions were applied, as well as the Gaussian one, in both our technique and probabilistic relaxation labeling.

5. Experiments and Results

We have compared our algorithm with Probabilistic Relaxation Labeling (PRL) [3] in two sets of experiments.

In both sets of experiments, we generated codomain patterns with S points at random positions (but not coincident) in images of size 256x256. Then, we extracted randomly a subset of T of these points to build the template pattern. In all the experiments, the number of iterations for PRL was set as 200, since empirical evaluation has shown that convergence always occurs below this value.

In the first set of experiments, we assume that there is a small amount of noise in the domain pattern. The noise consists in adding independent random numbers drawn from a normal distribution with zero mean and varying standard deviation (std) to both the x and y coordinates of each point in the domain pattern. The value of std was set in 2 pixels. We then vary the size of the codomain pattern for a fixed size of the domain pattern. A set of 8 increasingly complex matching tasks were carried out, where graphs of size (T,S) = (10,15), (10,20), (10,25), (10,30), (10,35), (10,40), (10,45) and (10,50) were matched using both our method (which we denote by JT) and PRL. For each of these matching tasks, we performed 1000 runs and recorded the average result. This entire procedure was repeated for each type of point proximity function: Gaussian (G), Hyperbolic Tangent (H) and Increasing Weighting (I). The obtained performances are shown in Figure 2, where $(a) \dots (d)$ correspond to results for different values of the parameter σ in each type of proximity function. σ is such that the maximal possible value for $||y_t^{ij} - y_s^{kl}||$ is normalized to K, whose values are in the top of the images in Figures 2 and 3. This normalization prevents underflow.

In the second set of experiments, the sizes of both point patterns are kept constant. We used T = 10 and S = 20. The reason for using small values is to present a fair comparison: as we will see, the performance of PRL is only comparable to our method for small sizes of point patterns. For patterns with too many points, our method is by far more effective. We vary, in this set of experiments, the noise in the domain pattern. The standard deviation was progressively set to $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ pixels. Again, all this procedure was repeated for each type of point proximity matrix and, similarly to the previous experiment, for the same values of K. The final performance is presented in Figure 3.

6. Discussion

Figure 2 shows that, for a fixed noise level, the performance of JT is weakly sensitive to the complexity of the matching task. As S grows, JT outperforms PRL in all proximity functions. This is a very important result, since in many applications the size of the underlying graphs is large. As we mentioned earlier, PRL shows good performance for small graph sizes. However, as the graph gets larger, its performance is noticeably damaged, no matter which value of K is assumed. This fact restricts the usability of PRL for small matching problems, as has already been argued [4]. It is worth mentioning the fairly good performances of PRL-I for high K and of PRL-H for low K, what encourages the use of these alternative proximity functions instead of the Gaussian one when using PRL.

In Figure 3, one notice that the performance of the JT method under varying noise conditions is fairly insensitive to the value of K, being JT-G and JT-H equally effective, and JT-I poorer. The performance of PRL, on the other hand, shows to be sensitive to K, where the most startling behavior is due to PRL-H for low values of K, when PRL-H and JT-H (or JT-G) perform similarly. In this case, the advantage of using the Hyperbolic function for PRL, instead of the Gaussian, is really significant.

In general, we observe that the main advantage of our method lies in its use for matching large graphs. In addition, the performance of the Gaussian and the Hyperbolic proximity functions are comparable in our method.

7. Conclusion

This work has presented an analysis of the robustness of point matching in a previously proposed technique under different types of point proximity functions. By accomplishing a set of controlled experiments, we showed that the performance of our technique exceeds significantly that of probabilistic relaxation labeling for matching problems involving large graphs, irrespectively of the type of proximity function. For simpler problems with small graphs, our technique still performs better for two types of proximity functions, while for the third type the two methods are comparable in performance.

References

[1] T. S. Caetano, T. Caelli, and D. A. C. Barone. An optimal probabilistic graphical model for point set matching. *University of Alberta, Dept. of Computing Science. Technical Report TR04-03*, 2004.

[2] M. Carcassoni and E. R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36:193–204, 2003.

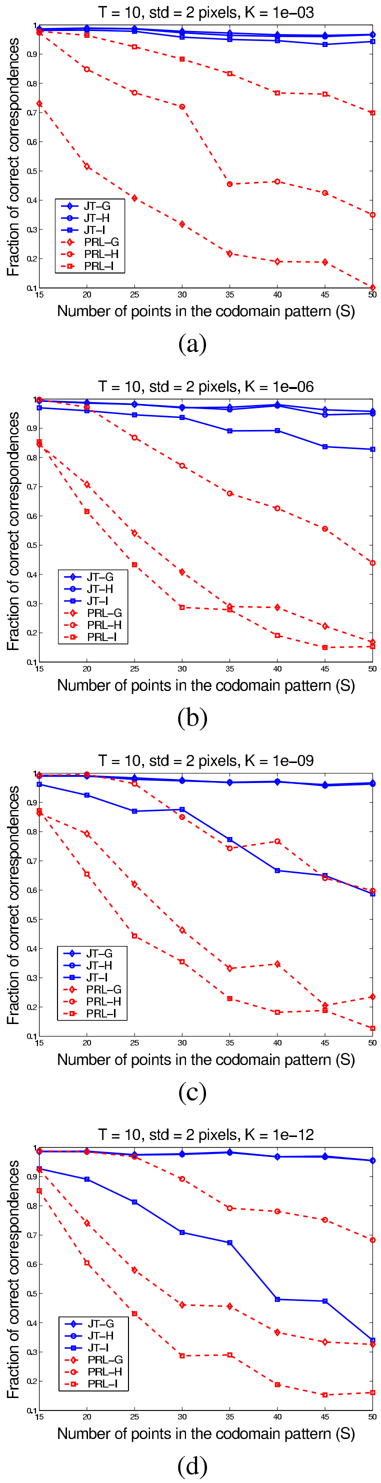
[3] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. PAMI*, 17(8):749–764, 1994.

[4] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. PAMI*, 18(4):377–388, 1996.

[5] M. I. Jordan. *An Introduction to Probabilistic Graphical Models*. MIT Press, Cambridge, MA, Forthcoming.

[6] S. L. Lauritzen. *Graphical Models*. Oxford University Press, New York, NY, 1996.

[7] D. Reimann and H. Haken. Stereo vision by self-organization. *Biological Cybernetics*, 71(1):17–26, 1994.



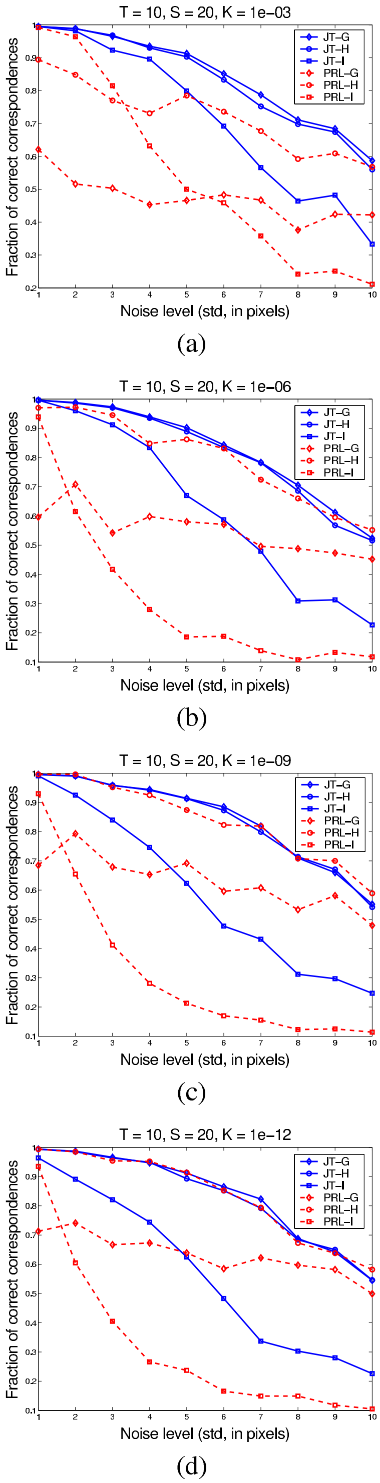
(a)

(b)

(c)

(d)

Figure 2. Robustness with respect to the size of the codomain pattern.



(a)

(b)

(c)

(d)

Figure 3. Robustness with respect to the noise level in the domain pattern.