Knowledge-Based Systems 90 (2015) 49–57

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Online learning the consensus of multiple correspondences between sets

Carlos Francisco Moreno-García, Francesc Serratosa*

Universitat Rovirai Virgili, Departament d'Enginyeria Informàtica i Matemàtiques, Av. Països Catalans 26, 43007 Tarragona, Spain

ARTICLE INFO

Article history:
Received 21 June 2015
Revised 26 September 2015
Accepted 28 September 2015
Available online 8 October 2015

Keywords:
Consensus
Learning weights
Correspondence between sets
Linear solver
Hamming distance

ABSTRACT

When several subjects solve the assignment problem of two sets, differences on the correspondences computed by these subjects may occur. These differences appear due to several factors. For example, one of the subjects may give more importance to some of the elements' attributes than another subject. Another factor could be that the assignment problem is computed through a suboptimal algorithm and different non-optimal correspondences can appear. In this paper, we present a consensus methodology to deduct the consensus of several correspondences between two sets. Moreover, we also present an online learning algorithm to deduct some weights that gauge the impact of each initial correspondence on the consensus. In the experimental section, we show the evolution of these parameters together with the evolution of the consensus accuracy. We observe that there is a clear dependence of the learned weights with respect to the quality of the initial correspondences. Moreover, we also observe that in the first iterations of the learning algorithm, the consensus accuracy drastically increases and then stabilises.

1. Introduction

Suppose we have several correspondences between sets and there is some level of intersection between them (Fig. 1 left). The aim of this paper is twofold. On the one hand, we define a method that enounces a consensus correspondence between these sets (Fig. 1 right). On the other hand, we present an online learning algorithm to set the meta-parameters needed to find this consensus correspondence. We face two main problems while seeking the consensus correspondence. First, there are discrepancies between the elements' mappings. Second, the intersection between sets is not null, although some elements may belong to only one or few sets. Fig. 1 schematically shows the consensus method. In this case, we suppose there are three different correspondences f^1, f^2 and f^3 that map their pairs of sets, and the intersection of sets is not null. Our method deducts A and A', as well as the consensus correspondence f.

In a real application, discrepancies between correspondences appear due to several factors. For example, one of the strategies may give more importance to some of the element's attributes, while the other strategy may believe another attribute is more important. If our scenario is based on an automatic method, these differences are gauged by the features or the weights of these features. Contrarily, if the scenario is based on a human-machine interaction (for example, semi-automatic medical image recognition), the strategy is based on the experience of a human specialist. If such elements in the sets represent regions of segmented images, one subject may think the area is more important than the colour, and the other one may think the opposite. Another factor that influences the elements' mapping happens when the assignment problem is computed with a suboptimal algorithm, and different non-optimal correspondences appear.

Some examples of methods that automatically solve the linear assignment problem are [1] or [2]. These methods return a bijective correspondence and the sets to be mapped have to be of the same order. There are other methods that this restriction is not needed and are the ones that discard outlier elements [3]. Finally, there are the ones that characterise the set of elements into an attributed graph [4–9]. Some methods have been presented to learn the graph-matching parameters [10,11]. Although some manual methods [12] have been presented to improve the correspondences made by a single matching algorithm, for these three scenarios, a consensus system could intervene as a third party to decide the final elements' correspondence when discrepancies appear, especially as the number of involved elements increase.

The rest of the paper is as follows. In Section 2, we review the methods related on finding a correspondence consensus. In Section 3, we present the basic definitions. In Sections 4 and 5, we explain the multiple-correspondence consensus method and we show the algorithm to learn the meta-parameters of the consensus method. In Section 6, we show the experimental validation and in Section 7, we conclude the paper.

* Corresponding author. Tel.: +34 636281015; fax: +34 977558710.
E-mail addresses: carlosfrancisco.moreno@estudiants.urv.cat (C.F. Moreno-García), francesc.serratosa@urv.cat, francesc.serratosa@gmail.com (F. Serratosa).

http://dx.doi.org/10.1016/j.knosys.2015.09.034
0950-7051/

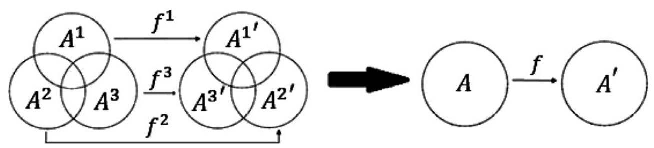

Fig. 1. Input of our problem: some correspondences between partially disjoin sets. Output: only one correspondence between two sets.

Note that in the experimental section of this paper, we apply our method to deduct a final correspondence such that its accuracy is better than the original correspondences between salient points extracted from images. Nevertheless, the method we present does not have to be seen solely as an image registration method, but as a method to deduct a new correspondence with better quality than the initial correspondences, given some sets of elements and such initial correspondences between them. Since the used databases are composed of images and the homographies between them, we can easily deduct the correct position of the salient points in the transformed image and create a ground truth correspondence.

2. Literature review

To the best of our knowledge, we are the first ones to tackle the problem of finding a consensus correspondence given a set of correspondences. We first analysed this problem considering only two correspondences in [13] (there is a preliminary version in [14]). Thus, we defined the consensus as the mean correspondence between both correspondences. The concept of mean was established through the Hamming distance between correspondences. The consensus correspondence is the one that obtains the same Hamming distance between it and both input correspondences. But at that point, we realised that the definition of the mean is an ill posed problem since there are several correspondences that hold this condition. We decided to return as the consensus the mean correspondence with the minimum cost since we assume the input correspondences tend to minimise some cost function. In [15], we formulated the methodology in [13] to be used on correspondences between attributed graphs. The main difference was the introduction of the second order costs defined on the graph edges. These costs influence on the cost function given a correspondence between two attributed graphs.

In [16] and [39], we generalise the problem and we presented two methods to deduct a correspondence consensus given several correspondences. The fact of increasing the number of correspondences involved in the process not only derives in an increase of the computational demand, but also an increase of the complexity of the problem at hand. In that paper, we proposed two different alternatives.

The first one is based on a voting process using the same technique such as [17]. In this case, each vote is an element-to-element mapping given a specific correspondence. The consensus correspondence is generated as follows. First, each possible element-to-element mapping accumulates all possible votes of the whole correspondences. Second, the element-to-element mappings are ordered given their votes. Third, the consensus correspondence is composed of the element-to-element mappings with the most votes that are congruent (they generate a bijective function). The second one is an incremental method. The algorithm sequentially executes the two-correspondence method presented in [14] and [13].

The contribution of the current paper with respect to [16] is twofold. First, we propose a general method to find the consensus given several correspondences based on a minimisation of an energy function, which is not based on the aforementioned voting method or iterative method. The main difference is that the function to be minimised considers the whole correspondences at the same time. Second, we define an algorithm to learn the contribution of each correspondence, that is, how much we believe on each correspondence.

Note that the algorithm we present and the ones in [16] (voting and iterative) obtain a consensus correspondence in a sub-optimal way. This is because the computational cost of an optimal algorithm is exponential with respect to the number and also order of sets, and therefore, seeking the optimal consensus is too computationally demanding in a real application.

Finally, in [18], authors deduct a consensus distance given several distances obtained from the same two images but using different features. Although the solution is applied for fingerprint matching, authors claim it can be easily extended to other type of images and features. The most important difference of this method and ours is that the inputs are some initial global distances and not some initial correspondences. Other interesting papers have been published related on the idea of generating a consensus given several data. For instance, in [20], a trust consensus is achieved given some social network analysis. In [19] and [21], a consensus decision is taken given some decisions of a set of people. In the second reference, authors apply fuzzy techniques. And also, in they [39] define consensus measures through fussy techniques.

3. Basic definitions and methods

In this section, we present three basic definitions. (1) The mean of a set of elements given any domain of the involved elements, (2) the distance between two sets considering outlier rejection and (3) the mean correspondence given a set of correspondences.

3.1. Set of elements and mean of a set of elements

Suppose we have a set of elements $A = \{a_1, \ldots, a_n,\}$ on the domain $a_i \in$ T. The mean $\bar{a} \in$ T of the elements in A is defined as,

$$\bar{a} = \underset{\forall a \in \mathrm{T}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} distance(a, a_i) \right\} \tag{1}$$

being *distance* any distance measure defined on the domain T of these elements. This function can be minimised using optimal or sub-optimal minimisation methods depending on several features, such as the definition of the distance function or the dimension of the problem.

3.2. Distance between sets and correspondence between elements

Given two sets of elements and a correspondence between them, we say that the inlier elements are the elements on both sets that are mapped by the correspondence, and the outlier elements are the elements that are not. Since both sets can have different cardinality, the number of inliers and outliers in both sets can be different. To formalise this situation, it is usual to consider some extra elements in both sets, which are usually called null elements. Thus, the elements in the set have to be considered outliers if are mapped to null elements in the codomain set. In the same way, the elements in the codomain set have to be considered outliers if their argument value is a null element. From now on, we consider that given two sets and a correspondence between them, both sets have the same order and the correspondence is bijective.

More formally, if we have two sets of elements $A = \{a_1, \ldots, a_n, a_{n+1}, \ldots, a_{n+m}\}$ and $A' = \{a'_1, \ldots, a'_m, a'_{m+1}, \ldots, a'_{n+m}\}$ with order $n+m$, the first n elements of A are original elements and the m remaining elements are null elements. The attribute of null elements is not in T, and it is represented by symbol ε. Then, $a_{n+1} \in \varepsilon, \ldots, a_{n+m} \in \varepsilon$. Similarly, this holds for the first m elements in A′ and the n remaining elements of A′. Therefore, $a_{m+1} \in \varepsilon, \ldots, a_{n+m} \in \varepsilon$. Moreover, there is a bijective correspondence $f(a_i) = a'_j$ that maps elements of both sets. We define the