Pattern Recognition Letters 000 (2016) 1–10

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Optimum-Path Forest based on *k*-connectivity: Theory and applications☆

João Paulo Papa[a,*], Silas Evandro Nachif Fernandes[b], Alexandre Xavier Falcão[c]

[a] Department of Computing, São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bauru, SP 17033-360, Brazil
[b] Department of Computing, Federal University of São Carlos, Rod. Washington Luís, Km 235, São Carlos, SP 13565-905, Brazil
[c] Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, Campinas, SP 13083-852, Brazil

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Pattern classification
Optimum-Path Forest
Supervised learning

ABSTRACT

Graph-based pattern recognition techniques have been in the spotlight for many years, since there is a constant need for faster and more effective approaches. Among them, the Optimum-Path Forest (OPF) framework has gained considerable attention in the last years, mainly due to the promising results obtained by OPF-based classifiers, which range from unsupervised, semi-supervised and supervised learning. In this paper, we consider a deeper theoretical explanation concerning the supervised OPF classifier with *k*-neighborhood (OPF_k), as well as we proposed two different training and classification algorithms that allow OPF_k to work faster. The experimental validation against standard OPF and Support Vector Machines also validates the robustness of OPF_k in real and synthetic datasets.

1. Introduction

Roughly speaking, pattern recognition techniques aim at learning a function that maps the input data to a set of predicted labels or continuous-valued outputs. Depending on the amount of knowledge we have about the training set, we can classify pattern recognition techniques in two main approaches: (i) supervised learning, which refers to situations one has full knowledge about the training data, and (ii) unsupervised learning, where we have no information about the dataset [7]. Recently, a new sort of approaches have been referred to semi-supervised ones, which feature some knowledge about a small subset of the training data. Such approaches make use of the active learning theory, which aims at improving data classification by means of user interaction.

Cutting edge research on pattern recognition may have contributed with its prominent works in the last years. Advances in hardware technology have allowed complex mathematical theories to be in lockstep with machine learning-based software development. Probabilistic models, techniques based on statistical learning theory and neural networks have been always the forerunners for pattern recognition-like applications. Support Vector Machines (SVM) [6], for instance, may be considered the hallmark with respect to kernel-based learning techniques. Since we can face complex and overlapped feature spaces, it might be interesting to employ kernel functions to map the data onto a higher dimensional representation.

Neural networks still play an important hole in the pattern recognition research field, since there is always room for improvements in the old fashion techniques [15,21,23]. In the last years, a special attention has been devoted to deep learning architectures [2,13], since they can be very robust to changes in scale, rotation and brightness in regard to image classification tasks. Recent advances in Bayesian networks [4,10], *k*-means [14] and the well-known Gaussian Mixture Models [5] have maintained the tradition of such techniques.

Another interesting framework that leads to a very interesting and powerful tool for pattern recognition concerns with graph-based methods. Basically, such methods model the machine learning task as a problem formulated in the graph theory: the dataset samples, which are represented by their corresponding feature vectors, are the graph nodes, that are further connected by an adjacency relation. Without loss of generality, a graph-based method aims at removing or adding edges using some heuristic in order to create connected components, which stand for a group of samples that share some similar characteristics [3].

Papa et al. [19,20] presented a new framework for graph-based pattern recognition named Optimum-Path Forest (OPF), which addresses the graph partition task as a competition process among some key (prototype) samples in order to conquer the remaining nodes according to a path-cost function. The idea is based on the Image Foresting Transform (IFT) [8], which works similarly to OPF,

☆ This paper has been recommended for acceptance by Cheng-Lin Liu.

* Corresponding author. Fax: +55 14 3103 6079.
E-mail address: papa@fc.unesp.br, papa.joaopaulo@gmail.com (J.P. Papa).

http://dx.doi.org/10.1016/j.patrec.2016.07.026
0167-8655/

but in the context of designing image processing-like operators. Both OPF and IFT follow the idea of the ordered communities formation, in which an individual (node) will belong to the community (cluster) that gives him/her the best reward (path-cost function value).

In order to create an OPF-based classifier, we have to face three main questions: (i) how to connect samples, (ii) how to find out prototypes, and (iii) what sort of path-cost function one should employ in the competition process. Papa et al. [19,20] and Papa and Falcão [17,18] have addressed the above questions in two distinct ideas: (i) using a complete graph as adjacency relation, a Minimum Spanning Tree (MST)-based approach to find out prototypes, and a path-cost function (f_{max}) that computes the maximum arc-weight along a path (sequence of nodes) [19,20]; and (ii) using a k-nearest neighbors (k-nn) adjacency relation, a density-based approach to estimate prototypes, and a path-cost function (f_{min}) that computes the minimum value between the cost of a training node and the density of the test sample [17,18]. This latter formulation is based on the unsupervised OPF [22], which was proposed to handle data clustering problems. Recently, Souza et al. [24] proposed a new variant called k-OPF, which essentially assigns the most frequent label of the k-lowest path-costs to a given sample, instead of the lowest path-cost only.

The main differences regarding the OPF with complete graph and its version that employs an adjacency relation based on k-connectivity (OPF_{knn}) rely on: (i) the naïve OPF weights only edges, while (OPF_{knn}) weights both edges and nodes; (ii) the prototypes estimated by OPF are located at the frontier of the classes, and the key nodes estimated by OPF_{knn} are positioned at the regions with highest density (center of clusters), and (iii) the classification process adopted by traditional OPF aims to *minimize* the cost of every sample using a path-cost function that computes the *maximum* arc-weight along a path; and the classification process of OPF_{knn} tries to *maximize* the cost of every sample using a path-cost function that computes the *minimum* value between the cost of a training sample and the density of a test node. Notice k-OPF and OPF_{knn} are different to each other, since the first one uses the complete graph as adjacency relation, it computes the prototypes using the Minimum Spanning Tree approach, and uses f_{max} as the path-cost function. The latter approach employs a k-nn graph, it computes prototypes based on a probability density function, and uses f_{min} as the path-cost function.

In this paper, we extend the research of Papa and Falcão [17,18] by addressing in more details the working mechanism of OPF_{knn}, as well as we propose to model the problem of finding the size of the k-neighborhood as an optimization task using meta-heuristics. Since Papa and Falcão [17] proposed to use an exhaustive search for finding the best value of k, i.e., the one that maximizes the accuracy over the training set, our approach can speed up the original work, as well as we can reduce the overtraining, since the proposed optimization process is conducted over a validating set. Another contribution of this work is to take advantage of the cost of each training sample, which has been computed during the training phase already, when classifying samples, i.e., we can simply halt the classification process earlier without affecting the theoretical basis of the algorithm. Therefore, the main contributions of this paper are three-fold: (i) to present a deeper formulation with respect to OPF_{knn}, (ii) to propose a meta-heuristic-based approach to automatically estimate the neighborhood size for density computation purposes, and (iii) to propose a faster classification process for the OPF_{knn} technique. In addition, we have compared OPF_{knn} against traditional OPF and Support Vector Machines.

The remainder of the paper is organized as follows. Sections 2 and 3 present the OPF_{knn} background theory and the proposed approach to speed up both the training and testing phases, respectively. Experiments are discussed in Section 4, and Section 5 states conclusions and future works.

2. Optimum-Path Forest with *knn*-connectivity

In this section, we describe the theory related to OPF_{knn}, as well as the basis of OPF-based classifiers.

2.1. Theoretical background

Let $\mathcal{Z}$ be a labeled dataset such that $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2 \cup \mathcal{Z}_3$, where $\mathcal{Z}_1$, $\mathcal{Z}_2$ and $\mathcal{Z}_3$ stand for a training, validating and testing sets, respectively. A graph $G = (\mathcal{V}, \mathcal{A})$ can be derived from $\mathcal{Z}$ such that each $s \in \mathcal{Z}$ becomes a graph node $v(s) \in \mathcal{V}$, where $v(\cdot)$ stands for a function that extracts the feature vector of given dataset sample (e.g., image, pixel, voxel or signal). Additionally, $\mathcal{A}$ denotes an adjacency relation that connects the samples in $\mathcal{V}$, and $d : \mathcal{V} \times \mathcal{V} \to \Re^+$ defines a function that is used to weight the edges in $\mathcal{A}$. Analogously to the construction of G, we can also derive $G_1 = (\mathcal{V}_1, \mathcal{A}_1)$, $G_2 = (\mathcal{V}_2, \mathcal{A}_2)$ and $G_3 = (\mathcal{V}_3, \mathcal{A}_3)$ from $\mathcal{Z}_1$, $\mathcal{Z}_2$ and $\mathcal{Z}_3$, respectively. However, as the adjacency relation is the same for the entire dataset, we can adopt $\mathcal{A}$ for all graphs.

Let π_s be a path in $\mathcal{G}$ with terminus in node $s \in \mathcal{V}$, and $\langle \pi_s \cdot (s, t) \rangle$ be the concatenation between path π_s and the arc $(s, t) \in \mathcal{A}$. We also denote $\langle t \rangle$ as being a trivial path. The idea of an OPF-based classifier is to use a smooth path-cost function f in order to rule a competition process in G among a set of prototype nodes $\mathcal{S} \subseteq \mathcal{V}$. The OPF algorithm aims at minimizing/maximizing $f(s)$ for every sample $s \in \mathcal{V}$, being the smoothness of f defined as follows [8]: for every sample $t \in \mathcal{V}$, there exists an optimum-path π_t which is trivial or can be represented by $\langle \pi_s \cdot (s, t) \rangle$, where

- $f(\pi_s) \leq f(\pi_t)$;
- π_s is optimum; and
- for every optimum-path τ_s, $f(\langle \tau_s \cdot (s, t) \rangle) = f(\pi_t)$.

The OPF proposed by Papa et al. [19,20] adopts $\mathcal{A}$ as a complete graph, the prototype set $\mathcal{S}$ is designed as being the connected samples in an MST computed over the training set, and f outputs the maximum arc-weight along a path (f_{max}). Such OPF configuration is motivated by the fact that an Optimum-Path Forest computed over a graph using f_{max} follows the shape of an MST computed over it, which means we can obtain the very same Optimum-Path Forest as previously computed using OPF by just removing the arcs that connect samples from different classes in the MST, and then propagating their costs using f_{max}. This behavior was observed by the work of Alléne et al. [1], and it has been used to make OPF training phase faster [11]. If one has an unique MST, i.e., all arc-weights are different to each other, the OPF classification error over the training set would be reduced to zero.

The main problem in reducing the error over the training set is related to a possible data overfitting. Therefore, motivated by such assumption, Papa and Falcão [17] proposed the OPF_{knn}, which models $\mathcal{A}$ as being an adjacency relation that connects each sample to its k-nearest neighbors (say that $\mathcal{A}_k$); the prototypes are now estimated as the nodes located at the highest density regions, and a path-cost function that aims at maximizing the cost of every sample is now employed. Roughly speaking, OPF_{knn} has two phases: a training and a classification step. The former is responsible for computing the density of each training node using $\mathcal{A}_{k^*}$, being k^* the best value of k that maximizes some criterion, and then to start the competition process among prototypes. After that, we have an Optimum-Path Forest computed over the training set, which will be used to classify each test sample. The classification process just picks up a sample from the test set, connects it to its k^*-nearest neighbors in the Optimum-Path Forest generated by the training

Please cite this article as: J.P. Papa et al., Optimum-Path Forest based on k-connectivity: Theory and applications, Pattern Recognition Letters (2016), http://dx.doi.org/10.1016/j.patrec.2016.07.026