Neurocomputing 149 (2015) 1299–1304

Contents lists available at ScienceDirect

Neurocomputing

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

Letters

Spectral clustering with the probabilistic cluster kernel

Emma Izquierdo-Verdiguier [a,*,1], Robert Jenssen [b], Luis Gómez-Chova [a,1], Gustavo Camps-Valls [a,1]

[a] Image Processing Laboratory (IPL) University of Valencia C/ Catedratico José Beltrán, 2. Paterna (Valencia), 46980, Spain
[b] University of Tromsø, Norway

ARTICLE INFO

Article history:
Received 17 June 2013
Received in revised form
17 August 2014
Accepted 26 August 2014
Communicated by D. Zhang
Available online 20 September 2014

Keywords:
Kernel methods
Generative kernels
Manifold learning
Spectral clustering

ABSTRACT

This letter introduces a *probabilistic cluster kernel* for data clustering. The proposed kernel is computed with the composition of dot products between the posterior probabilities obtained via GMM clustering. The kernel is directly learned from the data, is parameter-free, and captures the data manifold structure at different scales. The projections in the kernel space induced by this kernel are useful for general feature extraction purposes and are here exploited in spectral clustering with the canonical *k*-means. The kernel structure, informative content and optimality are studied. Analysis and performance are illustrated in several real datasets.

1. Introduction

Clustering is of fundamental importance in data analysis. This is reflected in the vast literature on the subject, including well-known methods such as *k*-means and Gaussian mixture models (GMMs) [1]. Recently, very promising approaches to clustering have been proposed in the form of the interrelated kernel-based and graph-spectral techniques [2–4]. These methods typically consist of two separate stages: first, features are generated based on the (top) eigenvalues (spectrum) and eigenvectors of a matrix that encodes similarities between pairs of data objects. Then, extracted features are globally clustered using *k*-means. The main advantages of such methods are their well-understood behavior in terms of linear algebra and their ability to correctly cluster both linear and nonlinear data structures.

The similarity (kernel) matrix is commonly computed based on a parameterized function such as the radial basis function (RBF). The most important parameter in RBFs is the width, which basically determines a fixed scale of analysis, and the choice of this parameter is of paramount importance. Lately some probabilistic approaches have been introduced to design kernel functions that capture the signal characteristics. Among them, we stand out the Fisher kernel [5] which combines the advantages of generative kernels with discriminative methods. Within generative approaches [6] exist different methodologies, such as exploiting the probabilistic nature of generative embeddings with information theoretical kernels [7] or kernels based on GMM [8]. Nevertheless, although all these kernel functions have shown very good results, three main shortcomings arise: (1) they all require first assuming a data generative model (e.g. Gaussian [8], Riccian [7], etc.) for which explicit metaparameter-dependent feature extractors need to be derived; (2) they have all been specifically designed and applied to supervised problems, mainly through the Support Vector Machine (SVM); and (3) they need *a priori* knowledge about the data to fix crucial parameters. These problems prevent using such kernels for data clustering, as no prior knowledge (besides the number of clusters in many cases) is assumed.

In this letter, we address all these issues by presenting a parameter-free kernel function based on clustering and used for data clustering. The idea is to encode similarity between objects using their probability of being grouped together at different scales, which is obtained from multiple "weak" learners based on GMM clustering. These local linear clusterings are then combined to build a global multiscale kernel that is used for spectral decomposition. As a result, an ensemble of linear clusterings enables nonlinear clustering [9].

The key quantity we introduce is a generative probabilistic cluster kernel function that is learned directly from the data by looking at

* Corresponding author.
E-mail addresses: emma.izquierdo@uv.es (E. Izquierdo-Verdiguier), robert.jenssen@uit.no (R. Jenssen), luis.gomez-chova@uv.es (L. Gómez-Chova), gustavo.camps@uv.es (G. Camps-Valls).

[1] This work is supported by the Spanish Ministry of Innovation and Science under Project TIN2012-38102-C03-01 (LIFE-VISION) and the Generalitat Valenciana under Project GV/2013/079.

http://dx.doi.org/10.1016/j.neucom.2014.08.068
0925-2312/

local-to-global similarities along the manifold. This entails no parameter tuning, which is especially beneficial in the current context of unsupervised clustering. We analyze main properties of the kernel and compare it to the standard RBF kernel and other kernel clustering approaches. The structure, informative content and optimality are studied. Analysis and performance are illustrated in several real problems.

2. Probabilistic cluster kernel (PCK)

Given n data points $\mathbf{x}_i \in \mathbb{R}^d$, $i=1,\ldots,n$, the proposed generative kernel, $K_c(\mathbf{x}_i, \mathbf{x}_j)$, is directly learned by clustering the available data. In particular, we assume a Gaussian mixture model (GMM) and apply the Expectation-Maximization (EM) algorithm to cluster the data. We repeat this operation for a different number of clusters, $g=2,\ldots,G+1$, and different initializations, $q=1,\ldots,Q$, resulting in $Q\times G$ clusterings (data partitions). Then, we calculate the membership of each sample $\mathbf{x}_i$ to the clusters, i.e. the posterior probability vector,[2] $\boldsymbol{\pi}_i(q,g) \in \mathbb{R}^g$, for each one of the estimated clusterings (see Algorithm 1). The *probabilistic cluster kernel* K_c is then computed as a composite kernel by averaging all the dot products between the posterior probability vectors [10]

$$K_c(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{Z} \sum_{q=1}^{Q} \sum_{g=2}^{G+1} \boldsymbol{\pi}_i(q,g)^\top \boldsymbol{\pi}_j(q,g), \tag{1}$$

where Z is a normalization factor. After building the kernel matrix with a sufficiently large number of clusters G and realizations Q, we proceed as in the standard spectral clustering approach, described above.

Algorithm 1. PCK kernel: training phase.

Require: $\{\mathbf{x}_i\}_{i=1}^n$: training data, Q: realizations, G: num. of clusters
Ensure: K_c: PCK kernel matrix, GMM clustering parameters: $\boldsymbol{\Theta}_{qg}$
  for $q=1$ to Q **do**
    for $g=2$ to $G+1$ **do** $\boldsymbol{\Theta}_{qg} \leftarrow$ EM-GMM with g clusters over $\{\mathbf{x}_i\}_{i=1}^n$
      for $i=1$ to n **do**
        $\boldsymbol{\pi}_i(q,g) \leftarrow P(\cdot|\mathbf{x}_i, \boldsymbol{\Theta}_{qg})$ GMM posteriors
        for $j=1$ to n **do**
          $\boldsymbol{\pi}_j(q,g) \leftarrow P(\cdot|\mathbf{x}_j, \boldsymbol{\Theta}_{qg})$ GMM posteriors
          $K_c(\mathbf{x}_i, \mathbf{x}_j) \leftarrow K_c(\mathbf{x}_i, \mathbf{x}_j) + \boldsymbol{\pi}_i(q,g)^\top \boldsymbol{\pi}_j(q,g)$
        end for
      end for
    end for
  end for

Algorithm 2. PCK kernel: test phase.

Require: $\{\mathbf{x}_i\}_{i=1}^n$: training data, $\{\mathbf{x}_j^*\}_{j=1}^m$: test data, GMM clustering parameters: $\boldsymbol{\Theta}_{qg}$
Ensure: K_c^*: PCK test kernel matrix
  for $q=1$ to Q **do**
    for $g=2$ to $G+1$ **do**
      for $i=1$ to n **do**
        $\boldsymbol{\pi}_i(q,g) \leftarrow P(\cdot|\mathbf{x}_i, \boldsymbol{\Theta}_{qg})$ GMM posteriors
        for $j=1$ to m **do**
          $\boldsymbol{\pi}_j^*(q,g) \leftarrow P(\cdot|\mathbf{x}_j^*, \boldsymbol{\Theta}_{qg})$ GMM posteriors
          $K_c^*(\mathbf{x}_i, \mathbf{x}_j^*) \leftarrow K_c^*(\mathbf{x}_i, \mathbf{x}_j^*) + \boldsymbol{\pi}_i(q,g)^\top \boldsymbol{\pi}_j^*(q,g)$
        end for
      end for
    end for
  end for

Intuitively, the probabilistic cluster kernel accounts for *probabilistic* similarities at small and large scales (which are related to the number of clusters, since a higher number of clusters implies local scales and *vice versa*) between all samples along the data manifold. On one hand, a high number of realizations Q improves the robustness of the ensemble of clusterings at the expense of increasing the computational cost. On the other hand, the optimum maximum number of clusters G should be (1) high to capture the local structure and greater than the number of desired classes in the dataset; and (2) reasonably lower than the number of samples (specially in high dimensional spaces) in order to estimate the data clusters accurately. Actually, the proposed kernel has a very important advantage that it does not assume an *ad hoc* parametric form or sophisticated priors and thus is more flexible and general. Moreover, the method does not require computationally demanding procedures. Note that whereas the cost of building the RBF kernel matrix is $\mathcal{O}(n^2)$, the PCK kernel involves both the estimation of the EM-GMM clustering and the kernel matrix generation for each one of the $Q\times G$ clusterings. However, the complexity of the eigen-decomposition of RBF or PCK kernels is $\mathcal{O}(n^3)$, which constitutes the real bottleneck of spectral clustering in large datasets. Finally, note that the proposed kernel generalizes previous (semi) supervised approaches based on cluster kernels, e.g. the approach in Weston et al. [11] is obtained wherein solely the cluster assignment with maximum posterior probability is considered. Moreover, it is worth noting that the proposed multiscale approach might also be applied to other generative kernels such as the Fisher kernel [5].

2.1. Properties

This subsection studies the main theoretical properties of the proposed cluster kernel in a Hilbert space.

Property 1. *The probabilistic cluster kernel performs a linear kernel in a posterior probability space.*

Proof. From Eq. (1), an arbitrary kernel function that forms the probabilistic cluster kernel is $K_c(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle = \langle \boldsymbol{\pi}_i, \boldsymbol{\pi}_j \rangle$, and then the explicit feature mapping is $\boldsymbol{\phi}(\mathbf{x}_i) = \boldsymbol{\pi}_i$. Therefore, the probabilistic cluster kernel computes second-order statistics in a probability space. □

Property 2. *The probabilistic cluster kernel K_c is a positive definite (p.d.) kernel.*

Proof. The function $K_c : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a p.d. kernel *if and only if* there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we have $K_c(\mathbf{x}, \mathbf{x}') = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}') \rangle_{\mathcal{H}}$. Using standard properties of kernel functions and Property 1, and as a simple consequence of the bilinearity of the dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, then $\forall c_i \in \mathbb{R}$:

$$\sum_{i,j=1}^{n} c_i c_j K_c(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i,j=1}^{n} c_i c_j \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle_{\mathcal{H}}$$

[2] The EM algorithm estimates the GMM parameters $\boldsymbol{\Theta}$ (prior α_k, mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ for each component k of the mixture) which are used to compute the posterior probabilities

$$P(k|\mathbf{x}_i, \boldsymbol{\Theta}) = \frac{\alpha_k |\boldsymbol{\Sigma}_k|^{-1/2} \exp(-(1/2)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k))}{\sum_l \alpha_l |\boldsymbol{\Sigma}_l|^{-1/2} \exp(-(1/2)(\mathbf{x}_i - \boldsymbol{\mu}_l)^\top \boldsymbol{\Sigma}_l^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_l))}.$$

In this work, the vector of posterior probabilities for the GMM obtained with g clusters and initialization q, $P(\cdot|\mathbf{x}_i, \boldsymbol{\Theta}_{qg})$, is referred to as $\boldsymbol{\pi}_i(q,g)$ for simplicity.