Fiche de lecture
Proposé par : Mme.Michèle Sebag
Réalisé par : Ounas ASFARI

Université de Paris sud – XI
M2R informatique
Cours: Apprentissage et Fouille de Données

Kernels on Structured Objects through Nested Histograms

Abstract

This article proposes a family of kernels for structured objects, the approach suggested here contains the following stages:
•decomposing each complex object into the single histogram of its components.
•use this hierarchy of histograms to define elementary kernels which can detect coarse and fine similarities between the objects.
•compute through an efficient averaging trick a mixture of such specific kernels, to propose a final kernel value which weights efficiently local and global matches.

1-Introduction:

•A kernel is intuitively a similarity measure between objects, It is thus a real-valued function of two variables on a given set of objects of interest X. (X can be a set of vectors describing features, graphs, strings, phylogenetic trees etc...)
• Kernel design aims at comparing complex objects in both a mathematical and application-motivated relevant way.
• Complex objects can often be decomposed into components in a set X.
A long sequence into n-grams:
AABHLKFHGH...HAABGJY HLKA.. ⟶ {(AAB, 2), (HLK, 2), (FHG, 1)...}

2- Kernels defined through Partitions

2.1 Creating Elementary Kernels from Local Similarities

• Suppose $\mathcal{L}$ is an index set, and consider families $\mu = \{\mu t\}_{t \in \mathcal{L}}$ of bounded measures μt of $M^b{+}(X)$ indexed on $\mathcal{L}$.
• Given an arbitrary kernel k on $M^b{+}(X)$ and a label $t \in \mathcal{L}$,

$$k_t(\mu, \mu') \stackrel{def}{=} k(\mu t, \mu' t)$$

Quantifies the similarity of μ and μ′ seen through label t.
• For two labels s and t, we can use k_s and k_t to define their sum $k_s + k_t$ as well as their product $k_s k_t$. On the other hand, if s and t are assumed to be close enough under a given granularity, then one can use a kernel built on the grouped labels {s} ∪ {t}

$$k_{\{s\} \cup \{t\}}(\mu, \mu') \stackrel{def}{=} k(\mu_s + \mu_t, \mu'_s + \mu'_t),$$

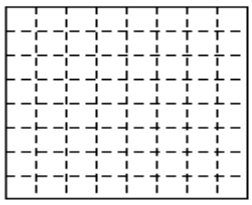
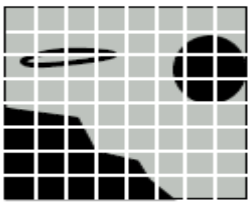
Which quantify how m is similar to m′ under both labels s and t.
• Natural extension to a set $T \subset \mathcal{L}$ of grouped labels, through

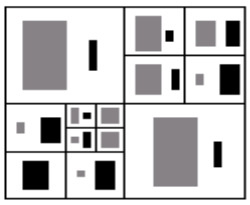
$$k_T(\mu, \mu') \stackrel{\text{def}}{=} k(\mu_T, \mu'_T), \text{where } \mu_T \stackrel{\text{def}}{=} \sum_{t \in T} \mu_t \text{ and } \mu'_T \stackrel{\text{def}}{=} \sum_{t \in T} \mu'_t.$$

2.2 Partition Specific Kernels

• Let P be a partition of $\mathcal{L}$, that is a finite family P = (T1, ..., Tn) of sets of $\mathcal{L}$, such that $T_i \cap T_j = \emptyset$ if $1 <= i < j <= n$ and $\cup_{i=1}^{n} T_i = \mathcal{L}$.
• Consider now the kernel defined by a partition P as

$$k_P(\mu, \mu') \stackrel{\text{def}}{=} \prod_{i=1}^{n} k_{T_i}(\mu, \mu')$$

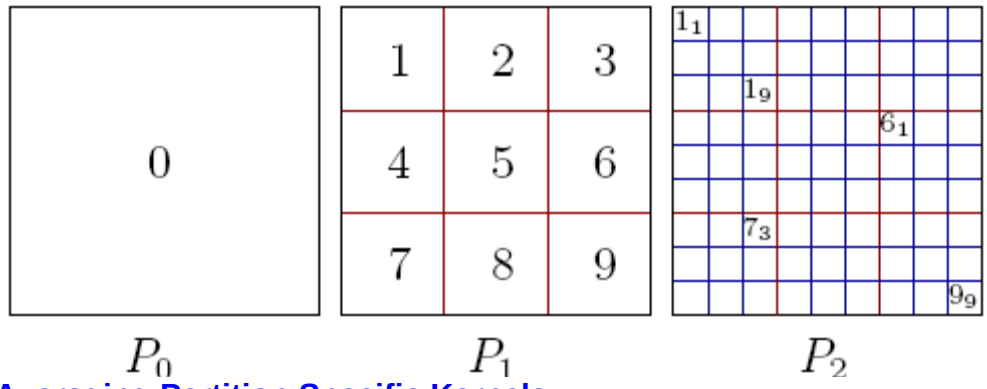





• Partitions with an increasing granularity can be obtained through a hierarchical structure on $\mathcal{L}$. A hierarchy is a family

$$(P_d)_{d=0}^{D} = \{P_0 = \{\mathcal{L}\}, .., P_D = \{\{t\}, t \in \mathcal{L}\}\}$$

of partitions, such that each subset T in P_d, is divided in P_{d+1} as a partition $S(T)$ of T which is not T itself.



2.3 Averaging Partition Specific Kernels

• Definition Let $\mathcal{L}$ be an index set endowed with a hierarchy $(P_d)_{d=0}^{D}$, π be a prior measure on the corresponding set of partitions P_D and k a base kernel on $M_+^b(\mathcal{X}) \times M_+^b(\mathcal{X})$. The **averaged kernel** k_π is defined as

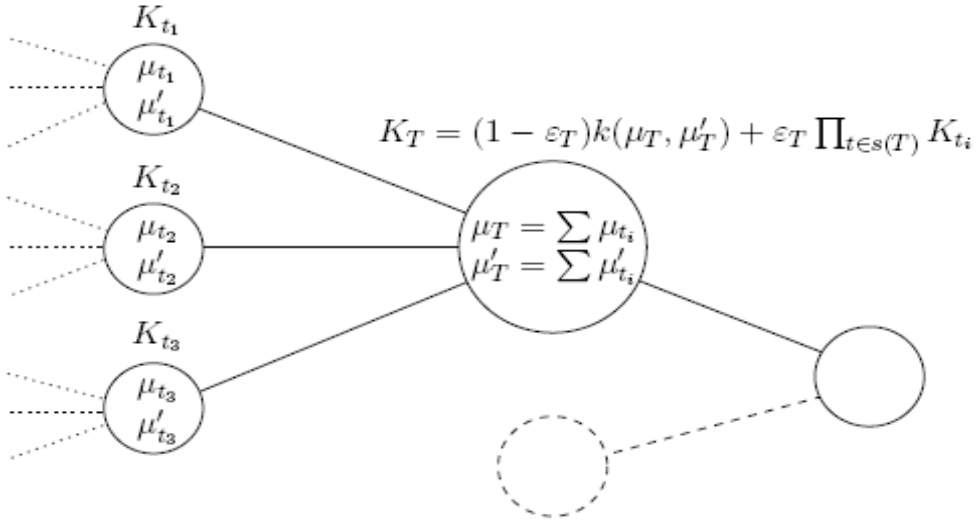
$$k_\pi(\mu, \mu') = \sum_{P \in \mathcal{P}_D} \pi(P)\, k_P(\mu, \mu')$$

3-Kernel Factorization:

• We can factorize Equation precedent when π is a branching process prior:
Partitions P of P_D are generated with a branching rule, starting from the root partition P: = P_0 = {$\mathcal{L}$} for each set T of P:

1. Either leave the set as it is in P with probability $1-\epsilon_T$,
2. Either replace it by its siblings in S(T) with probability ϵ_T , and reapply the rule to each sibling (unless they belong to P_D).

• Proposition For two elements μ, μ′ of M_+ (X), define for T spanning recursively from finest to coarsest all sets contained in P_D, P_{D-1}, ..., P_0 , the quantity K_T below;

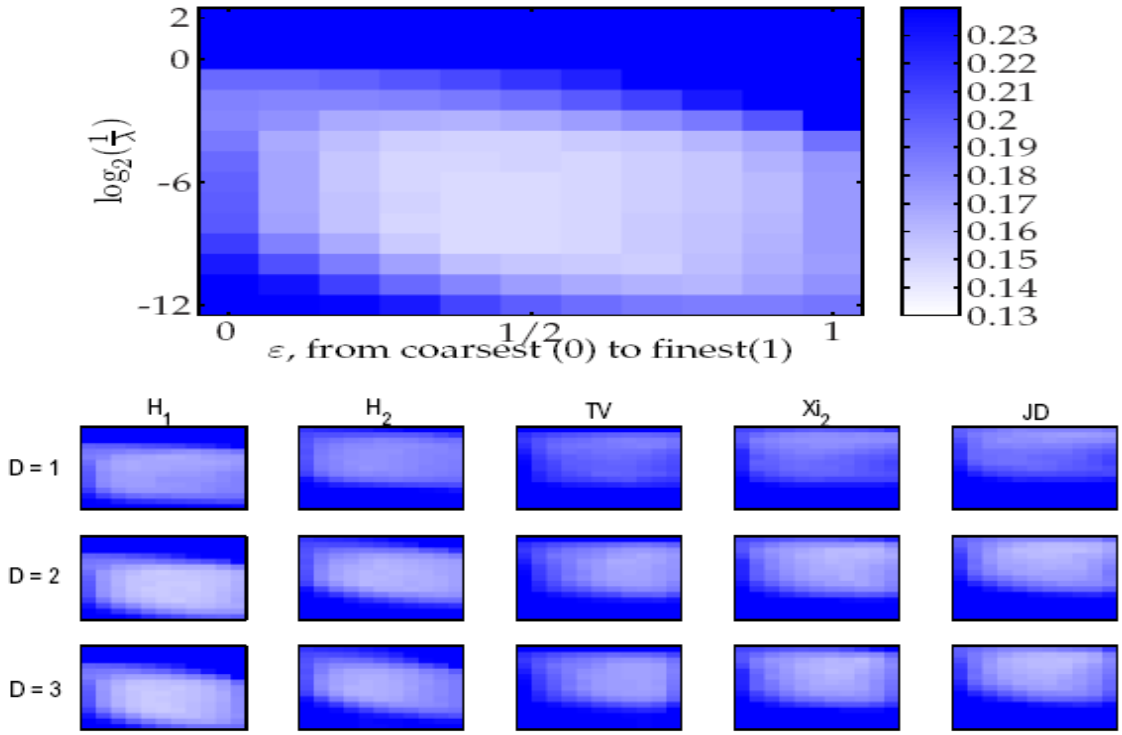


Then the averaged kernel $k_\pi(\mu, \mu')$ is retrieved through $K_{\mathcal{L}}$ (value at root node).

4-Experiments with Image Classification:

• We use 12 image families taken from the Corel database, each class containing 100 color images of 256 × 384 pixels,

• The DB is randomly split into balanced sets of 800 train images and 400 test images. The task consists in classifying the test images with the rule learned by training 12 one-versus-all SVM's on the learning fold. These results are averaged over 3 random splits (9 experiments in total), using the Spider toolbox.



5-Conclusions:

• The algorithm may be extended to kernels for vectors for which we can assume an hierarchical knowledge between coordinates.

• This framework is related to Multiple Kernel Learning, but here we use algebraic combinations rather than additive ones, and use a prior with no estimation of weights.

6-Comments and the answer of your question:

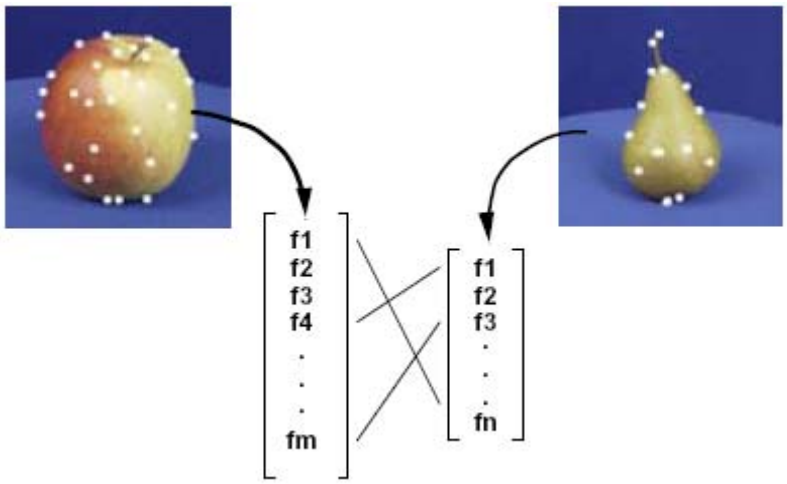
• Handling complex objects raises many questions: is "AAAA" more similar to "BBBB" or to "AAAAAAAAAAA"?
is "BABA" more similar to "DCDC "or to "AABB"?
Need to define meaningful similarity measures between complex objects.

• Defining a kernel is an arbitrary procedure; there is no universal kernel but rather useful families of kernels.

• Kernels methods work well with vectorial datapoints. Typically, kernels that take into account nonlinearities between the components provide good results.

The answer of your question:

Given two sets of local descriptors L1 and L2, at first a matrix of similarity scores between L1 and L2 is computed. Common choices for the similarity measure (the minor kernel) are the RBF-kernel:



$$k_{rbf}(f_1, f_2) = \exp\left(-\frac{\|f_1 - f_2\|^2}{2\sigma^2}\right)$$

Or the so-called normalized cross correlation kernel:

$$k_{normCC}(f_1, f_2) = \exp\left(-\rho\left(1 - \frac{(f_1 - \mu_1)\cdot(f_2 - \mu_2)}{\|f_1 - \mu_1\| \cdot \|f_2 - \mu_2\|}\right)\right)$$

Finally the kernel value is the average over the best-match-scores of the elements in L1 and L2:

$$K_{match}(\mathbf{L}_1, \mathbf{L}_2) = \frac{1}{2}\left[\hat{K}(\mathbf{L}_1, \mathbf{L}_2) + \hat{K}(\mathbf{L}_2, \mathbf{L}_1)\right]$$

With:

$$\hat{K}(\mathbf{L}_1, \mathbf{L}_2) = \frac{1}{|\mathbf{L}_1|}\sum_{i=1}^{|\mathbf{L}_1|} \max_{j=1\ldots|\mathbf{L}_2|} k(f_i, f_j)$$