Kernels and representation

Corso di AA, anno 2017/18, Padova

Fabio Aiolli

20 Dicembre 2017

Hierarchical Representation Learning

Recent research has addressed the representation problem by injecting some reasonable priors (regularization), including:

- Smoothness
- Multiple explanatory factors
- Shared factors across the tasks
- Manifolds
- Sparsity
- **Hierarchy of explanatory factors**

Hierarchical structure of factors with more *abstract* concepts/features higher in the hierarchy

Deep Neural Networks

Deep architectures

- Generate models with several levels of abstraction discovering more and more complicated structures;
- Current state-of-the-art in many different tasks;
- For instance, **Deep Neural Networks** (DNNs).

Some **drawbacks**:

- There is **not a clear decoupling** between the representation and the model generation;
- They have a **high training time** (tens of layers are difficult to be handled);
- They converge to a **sub-optimal solution** because of the local minima and the vanishing gradient issues.

Kernels and MKL

We consider the (implicit) representation given by **kernels**

A kernel can be seen as a **scalar product** in some Hilbert space, i.e $k(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$, where $\phi(\mathbf{x})$ is the **representation** for the example $\mathbf{x}$

Multiple Kernel Learning (MKL): Given a family $k_r(\mathbf{x}, \mathbf{z})$ of kernels such that

$$k_r(\mathbf{x}, \mathbf{z}) = \phi_r(\mathbf{x}) \cdot \phi_r(\mathbf{z}) \quad r = 1, \ldots, R$$

MKL optimizes the coefficients of a weighted sum of kernels: $\sum_{r=1}^{R} a_r k_r(\mathbf{x}, \mathbf{z}), a_r \geq 0$, hence computing a new implicit representation for $\mathbf{x}$ given by

$$\phi(\mathbf{x}) = [\sqrt{a_1}\phi_1(\mathbf{x}), \ldots, \sqrt{a_R}\phi_R(\mathbf{x})]^T$$

Expressiveness of kernel representations

The **expressiveness of a kernel** function, that is the number of **dichotomies** that can be realized by a linear separator in that feature space, is captured by the rank of the kernel matrices it produces.

Theorem

Let $\mathbf{K} \in \mathbb{R}^{L \times L}$ *be a kernel matrix over a set of* L *examples. Let* $rank(\mathbf{K})$ *be the rank of* $\mathbf{K}$*. Then, there exists at least one subset of examples of size* $rank(\mathbf{K})$ *that can be shattered by a linear function.*

Spectral Ratio

The **spectral ratio** (SR) for a positive semi-definite matrix $\mathbf{K}$ is defined as the ratio between the 1-norm and the 2-norm of its eigenvalues, or equivalently:

$$\mathcal{C}(\mathbf{K}) = \frac{||\mathbf{K}||_T}{||\mathbf{K}||_F}. \tag{1}$$

Note that, compared to the rank of a matrix, it does not require the decomposition of the matrix.

$$||\mathbf{K}||_T = \sum_i \mathbf{K}_{ii}$$

$$||\mathbf{K}||_F = \sqrt{\sum_{i,j} \mathbf{K}_{ij}^2}$$

Spectral ratio: properties

The (squared) spectral ratio can be seen as an (efficient) strict approximation of the rank of a matrix:

$$1 \leq \mathcal{C}(\mathbf{K}) \leq \sqrt{\text{rank}(\mathbf{K})}.$$

The spectral ratio $\mathcal{C}(\mathbf{K})$ has the following additional nice properties:

- the **identity** matrix has the **maximal** spectral ratio with $\mathcal{C}(\mathbf{I}_L) = \sqrt{L}$ (every possible 2^L dichotomies);
- the kernel $\mathbf{K} = \mathbf{1}_L \mathbf{1}_L^\top$, the constant matrix, has the **minimal** spectral ratio with $\mathcal{C}(\mathbf{1}_L \mathbf{1}_L^\top) = 1$ (only 2 dichotomies);
- it is **invariant to multiplication** with a positive scalar as $\mathcal{C}(\alpha \mathbf{K}) = \mathcal{C}(\mathbf{K}), \forall \alpha > 0$.

Hierarchical Structure of Kernel Representations

Definition

Let be given k_i, k_j, two kernel functions. We say that k_i *is more general than* k_j ($k_i \geq_G k_j$) whenever for any possible dataset $\mathbf{X}$, we have $\mathcal{C}(\mathbf{K}_{\mathbf{X}}^{(i)}) \leq \mathcal{C}(\mathbf{K}_{\mathbf{X}}^{(j)})$ with $\mathbf{K}_{\mathbf{X}}^{(i)}$ the kernel matrix evaluated on data $\mathbf{X}$ using the kernel function k_i.

The Proposed Framework

Given a hierarchical set of features F,

Learning over a hierarchy of feature spaces: the algorithm

1. Consider a partition $P = \{F_0, \ldots, F_R\}$ of the features and construct kernels associated to those sets of features, in such a way to obtain a set of kernels of increasing expressiveness, that is $k_0 \geq_G k_1 \geq_G \cdots \geq_G k_R$;
2. Apply a MKL algorithm on kernels $\{k_0, \cdots, k_R\}$ to learn the coefficients $\boldsymbol{\eta} \in \mathbb{R}_+^{R+1}$ and define $k_{MKL}(\mathbf{x}, \mathbf{z}) = \sum_{s=0}^{R} \eta_s k_s(\mathbf{x}, \mathbf{z})$.

The Proposed Framework

Several MKL methods to learn the weight vector η can be used, such as:

- based on margin optimization:
  - **SPG-GMKL** (Jain et al. 2012) very efficient and scalable to many base kernels
  - **EasyMKL** (Aiolli et al. 2015), very efficient and very scalable to many base kernels (based on KOMD)
  - ...
- based on radius-margin ratio optimization:
  - **R-MKL** (Do et al. 2009), which optimizes an upper bound of the radius margin ratio
  - **RM-GD** (Lauriola et al. 2017), able to optimize the exact ratio between the radius and the margin
  - ...

EasyMKL

EasyMKL (Aiolli and Donini, 2015) is able to combine sets of weak kernels by solving a simple quadratic optimization problem with

- Empirical effectiveness
- **High scalability** with respect to the number of kernels
  i.e. it is constant in memory and linear in time

Main idea: EasyMKL finds the coefficients of the MKL combination maximizing the distance (in feature space) between the convex hulls of positive and negative examples (**margin**).

The effectiveness strongly depends on the pre-defined weak kernels.

RM-GD

RM-GD (Lauriola, Polato and Aiolli, 2017) is a MKL algoritm able to combine kernels in a two-steps optimization process. Advantages w.r.t. other similar approaches include:

- **High scalability** with respect to the number of kernels (linear in both memory and time)
- **Sparsity** of the combination weights
- Optimization of the **exact** radius-margin ratio

Main idea: It exploits a gradient descent procedure, where at each step k

- It evaluates the current kernel by using the current combination weights $\boldsymbol{\eta}^{(k)}$
- It computes the gradient direction $\boldsymbol{g}^{(k)}$ and updates the combination weights $\boldsymbol{\eta}^{(k+1)} \leftarrow \boldsymbol{\eta}^{(k)} - \lambda \cdot \boldsymbol{g}^{(k)}$

The case of Dot-Product Kernels

Theorem

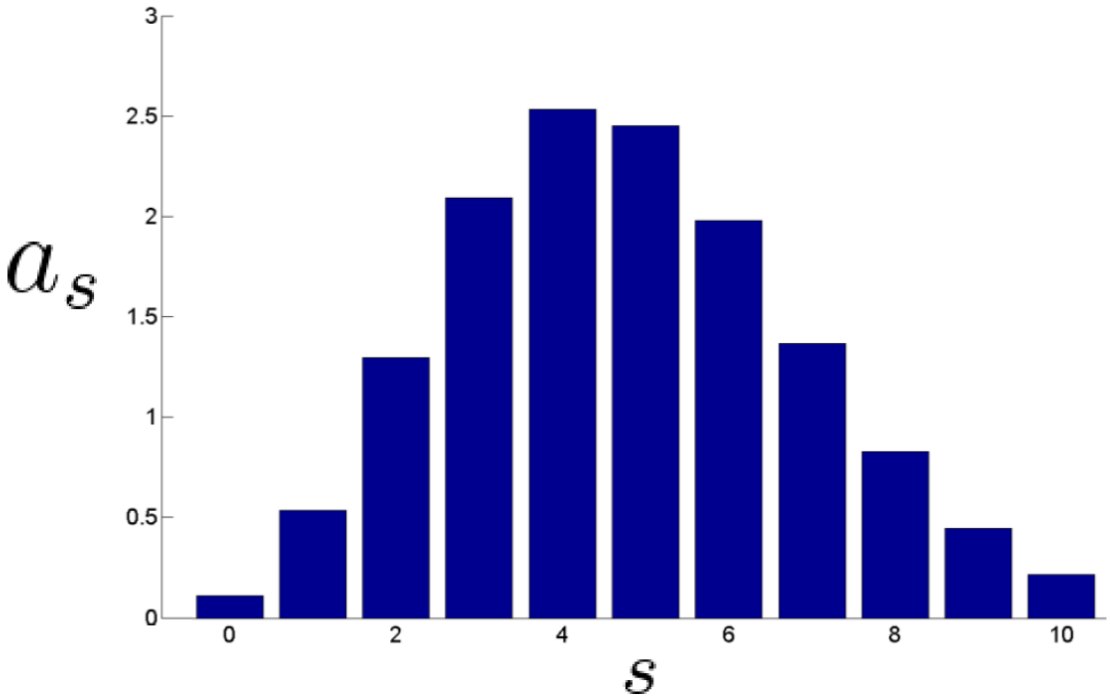
A function $f : \mathbb{R} \to \mathbb{R}$ defines a positive definite kernel $k : \mathbf{B}(\mathbf{0}, 1) \times \mathbf{B}(\mathbf{0}, 1) \to \mathbb{R}$ as $k : (\mathbf{x}, \mathbf{z}) \to f(\mathbf{x} \cdot \mathbf{z})$ iff f is an analytic function admitting a Maclaurin expansion with non-negative coefficients, $f(x) = \sum_{s=0}^{\infty} a_s x^s, a_s \geq 0$.

kernel	definition	DPP s-th coefficient (a_s)
Polynomial	$(\mathbf{x}^\top \mathbf{z} + c)^D$	$\binom{D}{s} c^{D-s}$
RBF	$e^{-\gamma \|\mathbf{x}-\mathbf{z}\|^2}$	$e^{-2\gamma} \frac{(2\gamma)^{2s}}{s!}$
Rational Quadratic	$1 - \frac{\|\mathbf{x}-\mathbf{z}\|^2}{\|\mathbf{x}-\mathbf{z}\|^2 + c}$	$\left(- \frac{2 \prod_{j=1}^{s} 2+(j-1)}{(2+c)^{s+1}} + \frac{\prod_{j=1}^{s} 2+(j-1)}{(2+c)^{s}} \right) \frac{1}{s!}$
Cauchy	$\left(1 + \frac{\|\mathbf{x}-\mathbf{z}\|^2}{\gamma}\right)^{-1}$	$\frac{s!!}{3^{s+1}\gamma^s} \frac{1}{s!}$

Table: DPP coefficients for some well-known dot-product kernels.

Famous examples: RBF

The DPP weights of the RBF kernels correspond to a parametric gaussian function. For example:



Homogeneus Polynomial Kernels

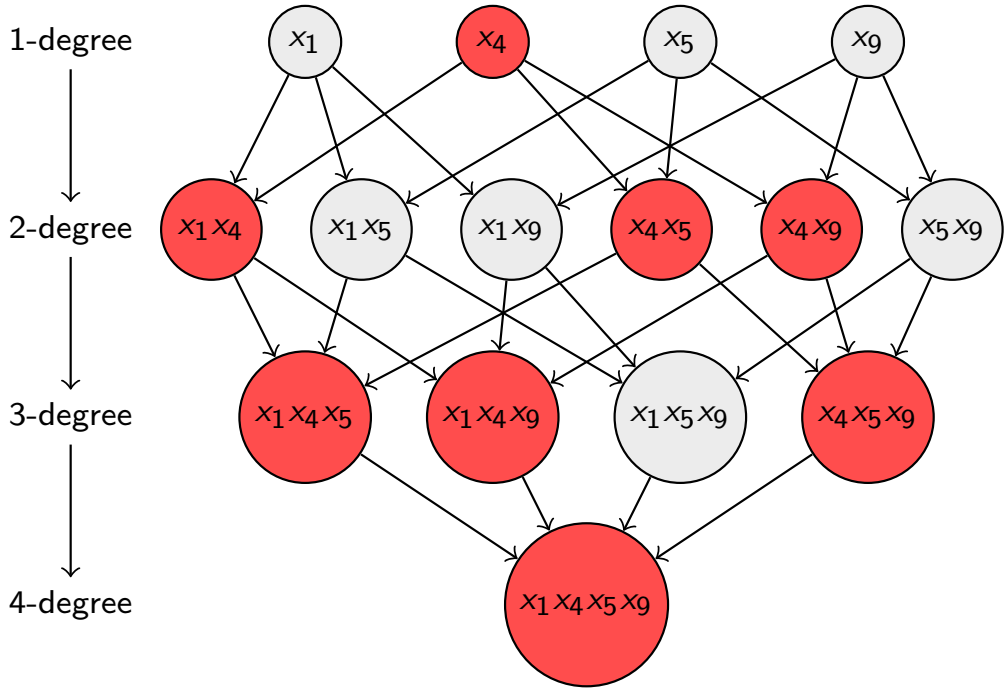


Figure: Arrows represent dependencies among features.

Exploiting these dependencies we were able to demonstrate that (when normalized) $k_s \geq_G k_{s+1}$ for any $s = 0, \ldots, D-1$

Non-parametric learning of DPK

The goal is to **learn the coefficients** a_s directly from data, thus generating a new dot-product kernel:

$$k(x, z) = \sum_{s=0}^{D} a_s (\mathbf{x} \cdot \mathbf{z})^s. \tag{2}$$

This problem can be easly formulated as a MKL problem where the weak kernels are HPKs defined as:

$$k_s(x, z) = (\mathbf{x} \cdot \mathbf{z})^s, \quad s = 0, \ldots, D, \tag{3}$$

for some fixed $D > 0$.

Dot-Product Kernels of Boolean vectors

For boolean vectors, similar expressiveness results can be given by using the conjunctive kernels in place of the homogeneous polynomial kernels.

Given $\mathbf{x}, \mathbf{z} \in \{0,1\}^n$, then any HP-kernel can be decomposed as a finite non-negative linear combination of C-kernels of the form:

$$\kappa^d_{HP}(\mathbf{x}, \mathbf{z}) = \sum_{s=0}^{d} h(s,d)\ \kappa^s_{\wedge}(\mathbf{x}, \mathbf{z}), \qquad h(s,d) \geq 0$$

Given $\mathbf{x}, \mathbf{z} \in \{0,1\}^n$ such that $\|\mathbf{x}\|_1 = \|\mathbf{z}\|_1 = m$, then any DPK can be decomposed as a finite non-negative linear combination of normalized C-kernels:

$$\kappa(\mathbf{x}, \mathbf{z}) = f(\langle \mathbf{x}, \mathbf{z} \rangle) = \sum_{s=0}^{m} g(m,s)\ \widetilde{\kappa}^s_{\wedge}(\mathbf{x}, \mathbf{z}), \qquad g(m,s) \geq 0$$

Empirical results

Average accuracy and ratio on binary datasets, by using different MKL algorithms

dataset	average	EasyMKL	RM-GD
audiology	$99.99_{\pm 0.04}$	$99.99_{\pm 0.04}$	$100.00_{\pm 0.00}$
(92,84,c)	$6.08_{\pm 0.33}$	$5.99_{\pm 0.32}$	$5.38_{\pm 0.25}$
primary-tumor	$72.55_{\pm 4.37}$	$72.69_{\pm 4.30}$	$74.58_{\pm 4.58}$
(132,24,c)	$15.87_{\pm 1.30}$	$15.05_{\pm 0.87}$	$14.31_{\pm 0.72}$
house-votes	$99.11_{\pm 0.41}$	$99.10_{\pm 0.42}$	$99.20_{\pm 0.41}$
(232,16,b)	$8.90_{\pm 1.13}$	$8.90_{\pm 1.17}$	$8.49_{\pm 1.13}$
spect	$82.01_{\pm 3.14}$	$82.06_{\pm 3.02}$	$83.39_{\pm 3.10}$
(267,23,b)	$18.91_{\pm 1.32}$	$18.56_{\pm 1.15}$	$17.53_{\pm 1.08}$
tic-tac-toe	$98.82_{\pm 0.46}$	$99.04_{\pm 0.39}$	$99.74_{\pm 0.20}$
(958,27,c)	$73.39_{\pm 1.57}$	$70.93_{\pm 1.45}$	$60.75_{\pm 1.49}$

Sparsity

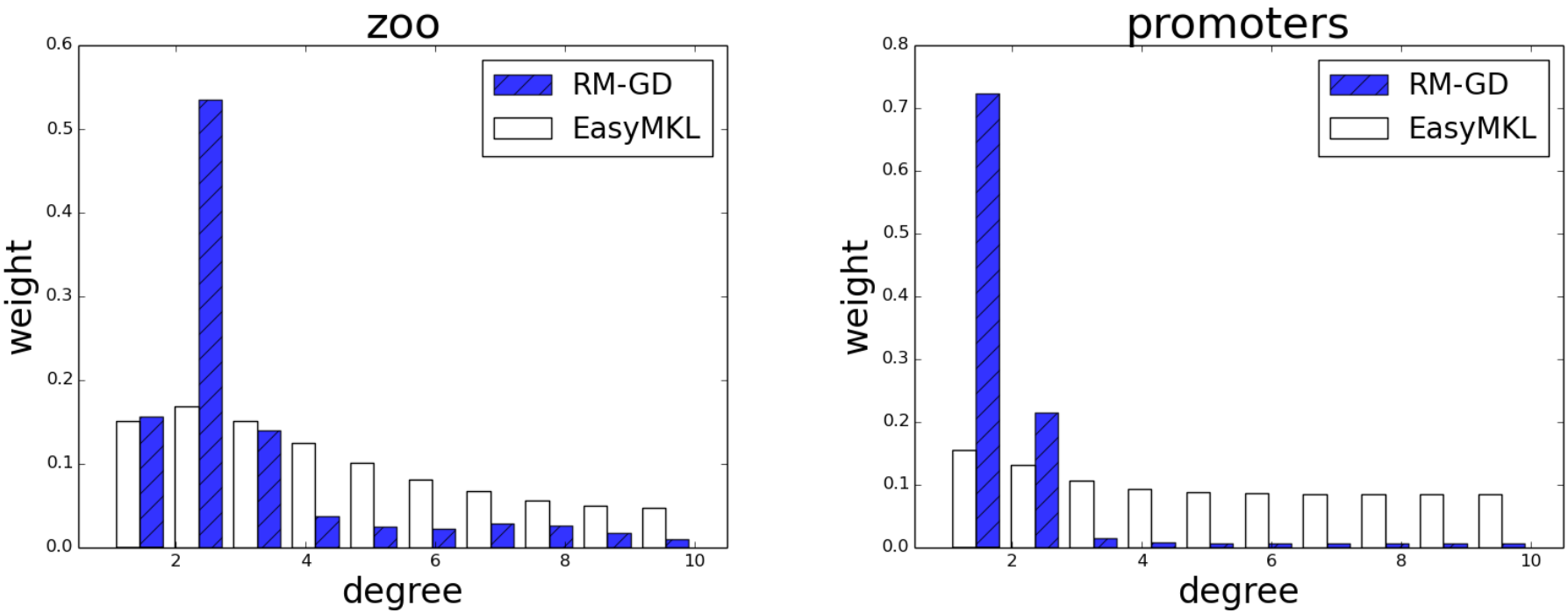


Figure: Combination weights learned when using 10 conjunctive kernels