Kernels and representation
Corso di AA, anno 2017/18, Padova

Fabio Aiolli

20 Dicembre 2017
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

\[ k(x, z) = \phi(x) \cdot \phi(z), \text{ where } \phi(x) \text{ is the representation for the example } x \]

**Multiple Kernel Learning (MKL):** Given a family \( k_r(x, z) \) of kernels such that

\[ k_r(x, z) = \phi_r(x) \cdot \phi_r(z) \quad r = 1, \ldots, R \]

MKL optimizes the coefficients of a weighted sum of kernels:

\[ \sum_{r=1}^{R} a_r k_r(x, z), a_r \geq 0, \text{ hence computing a new implicit representation for } x \text{ given by} \]

\[ \phi(x) = [\sqrt{a_1}\phi_1(x), \ldots, \sqrt{a_R}\phi_R(x)]^T \]
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 $\text{rank}(\mathbf{K})$ be the rank of $\mathbf{K}$. Then, there exists at least one subset of examples of size $\text{rank}(\mathbf{K})$ that can be shattered by a linear function.
Spectral Ratio

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

$$C(K) = \frac{\|K\|_T}{\|K\|_F}.$$  

(1)

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

\[
\|K\|_T = \sum_i K_{ii} \\
\|K\|_F = \sqrt{\sum_{i,j} K_{ij}^2}
\]
The (squared) spectral ratio can be seen as an (efficient) strict approximation of the rank of a matrix:

\[ 1 \leq C(K) \leq \sqrt{\text{rank}(K)}. \]

The spectral ratio \( C(K) \) has the following additional nice properties:

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

\[
C(K^{(i)}_X) \leq C(K^{(j)}_X)
\]

with \( K^{(i)}_X \) the kernel matrix evaluated on data \( X \) using the kernel function \( k_i \).
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 $\eta \in \mathbb{R}_+^{R+1}$ and define

$$k_{MKL}(x, z) = \sum_{s=0}^{R} \eta_s k_s(x, z).$$
The Proposed Framework

Several MKL methods to learn the weight vector $\eta$ 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 (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 (Lauriola, Polato and Aiolli, 2017) is a MKL algorithm 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 $\eta^{(k)}$
- It computes the gradient direction $g^{(k)}$ and updates the combination weights $\eta^{(k+1)} \leftarrow \eta^{(k)} - \lambda \cdot g^{(k)}$
The case of Dot-Product Kernels

**Theorem**

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

<table>
<thead>
<tr>
<th>kernel</th>
<th>definition</th>
<th>DPP s-th coefficient ( (a_s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial</td>
<td>((x^\top z + c)^D)</td>
<td>(\binom{D}{s} c^{D-s} )</td>
</tr>
<tr>
<td>RBF</td>
<td>(e^{-\gamma |x-z|^2})</td>
<td>(e^{-2\gamma (2\gamma)^{2s}} )</td>
</tr>
<tr>
<td>Rational Quadratic</td>
<td>(1 - \frac{|x-z|^2}{|x-z|^2+c})</td>
<td>(\frac{-2 \prod_{j=1}^{s} 2+(j-1) + \prod_{j=1}^{s} 2+(j-1)}{(2+c)^s+1} )  (\frac{1}{s!})</td>
</tr>
<tr>
<td>Cauchy</td>
<td>(\left(1 + \frac{|x-z|^2}{\gamma}\right)^{-1})</td>
<td>(\frac{s!!}{3^{s+1} \gamma^s} \frac{1}{s!})</td>
</tr>
</tbody>
</table>

**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:
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:

$$k(x, z) = \sum_{s=0}^{D} a_s (x \cdot z)^s.$$  \hspace{1cm} (2)

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

$$k_s(x, z) = (x \cdot z)^s, \quad s = 0, \ldots, D,$$ \hspace{1cm} (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 $x, z \in \{0, 1\}^n$, then any HP-kernel can be decomposed as a finite non-negative linear combination of C-kernels of the form:

$$
\kappa_{HP}^d(x, z) = \sum_{s=0}^{d} h(s, d) \kappa^s(x, z), \quad h(s, d) \geq 0
$$

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

$$
\kappa(x, z) = f(\langle x, z \rangle) = \sum_{s=0}^{m} g(m, s) \tilde{\kappa}^s(x, z), \quad g(m, s) \geq 0
$$
## Empirical results

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

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

Figure: Combination weights learned when using 10 conjunctive kernels