Lecture 15: Learning with Kernels

Prof. Dr. Mario Fritz

2022 06 13

https://fritz.cispa.saarland
Trustworthy AI
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Outline

Main references

- Learning with Kernels - Chapter 2
- Bishop - Chapter 6

Outline

Learning with kernels:

- As hypothesis space we use the RKHS $\mathcal{H}_k$ associated to the kernel k,
- As regularization functional we use: $\Omega(f) = \|f\|^2_{\mathcal{H}_k}$ (or more generally a strictly monotonically increasing function of $\|f\|_{\mathcal{H}_k}$)

Regularized empirical risk minimization problem with a RKHS as hypothesis space:

$$f^* = \arg\min_{f \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \Omega\Big(\|f\|^2_{\mathcal{H}_k} \Big),$$

Important observations

Problems

- The RKHS has often very high dimension or is even infinite dimensional. This means we have a very high dimensional hypothesis space.
- Thus, there is a danger of **overfitting**!

Solution:

- Regularization + **the representer theorem**!
- Effectively we are working in an n-dimensional subspace of $\mathcal{H}_k$!

Representer Theorem

Theorem (Representer Theorem)

Denote by $\Omega : [0, \infty) \to \mathbb{R}$ *a strictly monotonically increasing function. Let* $\mathcal{X}$ *be the input space,* $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ *an arbitrary loss function and* $\mathcal{H}_k$ *the reproducing kernel Hilbert space associated to the kernel* k*. Then, each minimizer* $f^* \in \mathcal{H}_k$ *of the regularized empirical risk*

$$f^* = \underset{f \in \mathcal{H}_k}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathrm{x}_i)) + \lambda \Omega\Big(\|f\|^2_{\mathcal{H}_k} \Big),$$

admits a representation as

$$f^*(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \mathrm{x})$$

Note also that $\|f^*\|^2_{\mathcal{H}_k} = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j)$.

Proof I

- $\mathcal{G} = \mathrm{Span}\{k(\mathrm{x}_i, \cdot) \mid i = 1, \ldots, n\}$ is the finite dimensional subspace of $\mathcal{H}_k$ spanned by the data.
- Decompose any $f \in \mathcal{H}_k$ into $f^{\parallel} \in \mathcal{G}$ and the orthogonal part $f^{\perp} \in \mathcal{G}^{\perp}$. Then,

$$f(\mathrm{x}) = f^{\parallel}(\mathrm{x}) + f^{\perp}(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \mathrm{x}) + f^{\perp}(\mathrm{x}).$$

- Note that since $k(\mathrm{x}_i, \cdot) \in \mathcal{G}$ and $f^{\perp} \in \mathcal{G}^{\perp}$ we have,

$$f^{\perp}(\mathrm{x}_i) = \left\langle f^{\perp}, k(\mathrm{x}_i, \cdot) \right\rangle_{\mathcal{H}_k} = 0,$$

  for all $i = 1, \ldots, n$. Therefore,

$$f(\mathrm{x}_j) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \mathrm{x}_j) + f^{\perp}(\mathrm{x}_j) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \mathrm{x}_j).$$

  Moreover,

$$\Omega\left(\|f\|^2_{\mathcal{H}_k} \right) = \Omega\left(\left\| f^{\parallel} \right\|^2_{\mathcal{H}_k} + \|f^{\perp}\|^2_{\mathcal{H}_k} \right) \geq \Omega\left(\left\| f^{\parallel} \right\|^2_{\mathcal{H}_k} \right)$$

Proof II

In words:

- Any function in the RKHS $\mathcal{H}_k$ decomposes as $f(\mathrm{x}) = f^{\parallel}(\mathrm{x}) + f^{\perp}(\mathrm{x})$.
- The training emprirical risk of any function $f(\mathrm{x})$ in $\mathcal{H}_k$ depends only on $f^{\parallel}(\mathrm{x})$.
- The regularizatiom term $\Omega\left(\|f\|^2_{\mathcal{H}_k}\right)$ is minimized when the optimal solution $f^*(\mathrm{x})$ can be written in terms of only $f^{\parallel}$.
- Thus, the solution to the regularized emprirical risk in the RKHS can always be written as:

$$f^*(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i k(\mathrm{x}_i, \mathrm{x}).$$

Kernelization of algorithms

When? I.e., **which learning methods can be used with kernels?**

- Any regularized empirical risk minimization problem of the form,

$$f^* = \underset{f \in \mathcal{H}_k}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \Omega\Big(\|f\|^2_{\mathcal{H}_k} \Big).$$

- Any method which can be formulated only using inner products (usually inner product in $\mathbb{R}^d$)

How? **Replace inner product with kernel, or equivalently, use the the representer theorem:**

- Final function: $f(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$.
- Regularizer: $\|f\|^2_{\mathcal{H}_k} = \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j)$.

Kernelization of algorithms II

- **Optimization point of view:** Transformation of any regularized empirical risk minimization problem of the form,

$$f^* = \arg\min_{f \in \mathcal{H}_k} \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, f(\mathrm{x}_i)\big) + \lambda \Omega\Big(\|f\|^2_{\mathcal{H}_k} \Big)$$

$$\Downarrow$$

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} L\Big(y_i, \sum_{j=1}^{n} \alpha_j k(\mathrm{x}_j, \mathrm{x}_i)\Big) + \lambda \Omega\Big(\sum_{i,j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j)\Big)$$

  and $f^*(\mathrm{x}) = \sum_{i=1}^{n} \alpha_i^* k(\mathrm{x}_i, \mathrm{x})$.
- **Geometric point of view:**
  - Map data to high-dimensional feature space: $\phi : \mathcal{X} \to \mathcal{H}_k$
  - Apply linear algorithm in $\mathcal{H}_k$. Equivalently, replace inner product with kernel function,

$$\langle \mathrm{x}, \mathrm{x}' \rangle_{\mathbb{R}^d} \quad \Longrightarrow \quad k(\mathrm{x}, \mathrm{x}') = \langle \Phi_{\mathrm{x}}, \Phi_{\mathrm{x}'} \rangle_{\mathcal{H}_k}.$$

General Scheme

Replace inner products with kernels:

- any linear method can be kernelized,
- often the dual formulation is more easily accessible and better suited for optimization,
- Kernel Logistic Regression, Kernel Fisher Discriminant Analysis, Kernel PCA, Kernel Perceptron, ...

Example: Regularized Least Squares/Ridge regression

$$f^* = \arg\min_{f \in \mathcal{H}_k} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2 + \lambda ||f||^2_{\mathcal{H}_k}$$

using representer theorem:

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{n}\alpha_j k(x_j, x_i))^2 + \lambda \sum_{i,j=1}^{n}\alpha_i\alpha_j k(x_i, x_j)$$

Kernelized regularized least squares/ridge regression in matrix/vector notation:

$$\arg\min_{\alpha \in \mathbb{R}^n} \frac{1}{n}\|Y - K\alpha\|^2 + \lambda\,\alpha^T K\alpha.$$

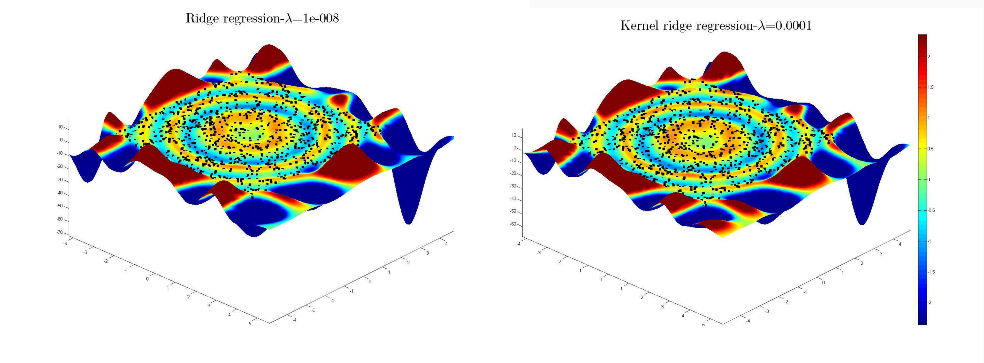
As stationary equation we get

$$K^T K\alpha + n\lambda\, K\alpha = K^T Y.$$

Assuming that K is invertible we get

$$\alpha = (n\lambda\mathbb{1} + K)^{-1} Y.$$

Example: Ridge versus Kernel ridge regression



- input: unif. on $[-\frac{7}{2}, \frac{7}{2}]^2$, output: $Y = \sin(\|X\|^2) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \frac{4}{100})$
- regularization parameter λ chosen by optimizing on test set,
- MSE for ridge regression was 0.121 and for kernel ridge regression 0.109,
- basis functions: $\phi_i(x) = e^{-\|x-x_i\|^2}$ and the Gaussian kernel, $\Longrightarrow$ solutions f^* have the expansion: $f^*(x) = \sum_{i=1}^{n} \alpha_i e^{-\|x-x_i\|^2}$,

Outline

The soft margin SVM is formulated using **slack variables** $\xi_i \geq 0$.

$$\min_{\mathrm{w}\in\mathbb{R}^d,\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^n} \frac{1}{2}\|\mathrm{w}\|_2^2 + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$\text{subject to: } y_i(\langle \mathrm{w}, \mathrm{x}_i\rangle + b) \geq 1 - \xi_i, \quad \forall\, i = 1, \dots, n, \qquad \xi_i \geq 0,$$

- the geometric margin is given by $\frac{2}{\|\mathrm{w}\|_2}$,
- maximizing the margin corresponds to minimizing $\|\mathrm{w}\|_2$,
- slack variables allow points to get inside the margin - soft margin

SVM = RERM with Hinge loss and squared regularizer:

$$\min_{\mathrm{w}\in\mathbb{R}^d,\ b\in\mathbb{R}} C\,\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1-y_i(\langle \mathrm{w}, \mathrm{x}_i\rangle + b)\right) + \|\mathrm{w}\|_2^2\,,$$

- error parameter C is inverse to the regularization parameter $\lambda = \frac{1}{C}$.

Dual problem:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \left\langle \mathrm{x}_i, \mathrm{x}_j\right\rangle\,,$$

$$\text{subject to: } 0 \le \alpha_i \le \frac{C}{n}, \quad i = 1,\ldots,n, \quad \sum_{i=1}^{n} y_i\alpha_i = 0.$$

Kernalized SVM

SVM = RERM with Hinge loss and squared regularizer:

$$\min_{f \in \mathcal{H}_k,\, b \in \mathbb{R}} C\,\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\langle \mathrm{w}, \phi(\mathrm{x}_i)\rangle + b)\right) + \|\mathrm{w}\|^2_{\mathcal{H}_k},$$

becomes with the representer theorem,

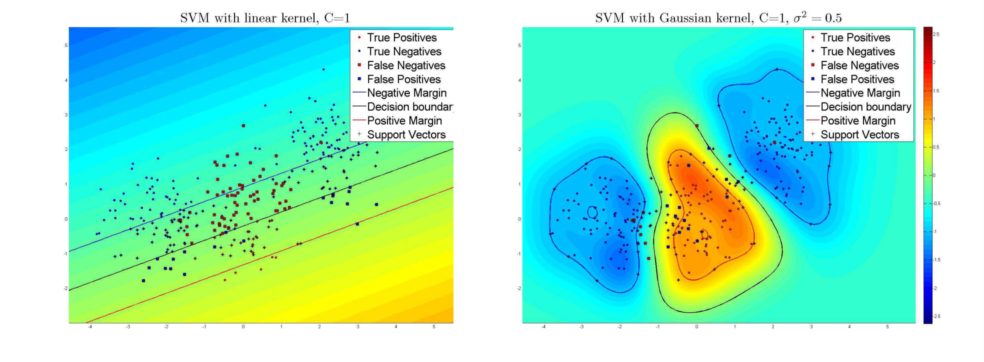
$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n,\, b \in \mathbb{R}} C\,\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i\Big(\sum_{j=1}^{n} \alpha_j k(\mathrm{x}_j, \mathrm{x}_i) + b\Big)\right) + \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\mathrm{x}_i, \mathrm{x}_j),$$

The dual problem:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j\, k(\mathrm{x}_i, \mathrm{x}_j),$$

$$\text{subject to: } 0 \leq \alpha_i \leq \frac{C}{n}, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} y_i \alpha_i = 0.$$

Example of Kernalized SVM



Left: the result of the linear SVM with error parameter C - clearly no linear hyperplane can solve this problem. **Right:** the result of the SVM with a Gaussian kernel with $\sigma^2 = \frac{1}{2}$ and $C = 1$. We observe that the Gaussian kernel can nicely identify the class structure.

(Image by Prof. Hein)

Outline

Regularization

What is the purpose of regularization?

- penalize functions which are not smooth, i.e., functions where small changes in the data lead to large changes in the prediction.
- regularization functional should measure complexity of the function.

How can we measure smoothness of a function?

- Penalize the derivatives of a function e.g. $\Omega(f) = \int_{\mathbb{R}^d} \|\nabla f\|_2^2 \, dx$.
- How can we achieve that using a RKHS? Can we see directly from a kernel what kind of regularization functional it induces?

Penalization of all derivatives:

The **Gaussian kernel**

$$k(x-y) = \exp\Big(-\frac{(x-y)^2}{2\sigma^2}\Big)$$

Thus we can argue (the rigorous mathematics is quite tricky (Bochner Theorem))

$$\|f\|^2_{\mathcal{H}_k} = \frac{\sigma}{\sqrt{2\pi}} \int_{\mathbb{R}} \sum_{j=0}^{\infty} \frac{\sigma^{2j}}{j!2^j} \Big(\frac{d^j f}{dx^j}\Big)^2 dx.$$

Regularization II

Translation invariant kernels in $\mathbb{R}^d$

$$k(x, y) = k(x - y).$$

What does translation invariant mean?

- *What?* Translating all feature vectors by a constant vector $c \in \mathbb{R}^d$, $x \mapsto x + c$, does not change the kernel.

$$k\big(x + c, y + c\big) = k\big((x + c) - (y + c)\big) = k\big(x + c - y - c\big) = k\big(x - y\big) = k\big(x, y\big).$$

- *When?* Use them if only **relative** properties of the features are important, but not **absolute** ones.

Translation and rotation invariant kernels

A **translation and rotation invariant kernel** has the form

$$k(x, y) = \phi(\|x - y\|^2).$$

Such kernels are called **radial**.

What means rotational invariance?

Let R be an orthogonal matrix, that is $RR^T = R^T R = \mathbb{1}$, then

$$\begin{aligned} k(Rx, Ry) &= \phi\big(\|Rx - Ry\|^2 \big) = \phi\big(\langle R(x-y), R(x-y) \rangle \big) \\ &= \phi\big(\big\langle (x-y), R^T R(x-y) \big\rangle \big) = \phi\big(\langle x-y, x-y \rangle \big) = \phi\big(\|x-y\|^2 \big) \\ &= k(x, y). \end{aligned}$$

Applying a rotation on the whole space does not change the kernel.

Translation and rotation invariant kernels II

Standard radial kernels:

$$\begin{aligned}
\text{Gaussian kernel:} \qquad k(x, y) &= \exp\left(- \frac{\|x - y\|^2}{2\sigma^2} \right), \\
\text{Laplace kernel:} \qquad k(x, y) &= \exp\Big(- \lambda \, \|x - y\| \Big).
\end{aligned}$$

Kernels can be defined on arbitrary sets !

Not any positive definite kernel is useful !

$$k(x,y) = c, \quad c \geq 0, \quad \forall\, x, y \in \mathcal{X},$$

$$k(x,y) = \begin{cases} 1 & \text{if} \quad x = y \\ 0 & \text{else} \end{cases} .$$

$\Longrightarrow$ no generalization possible.

Kernels on structured domains II

How we should we construct kernels (on structured domains) ?

- the kernel function $k(x, y)$ should be a natural similarity measure. In particular, objects

$$\text{for all } y \sim x \text{ then } k(x, y) \geq k(x, z) \text{ where } z \nsim x.$$

- distance function $d(x, y)$ induced by the kernel should be a natural dissimilarity measure.
- the evaluation of the kernel function should include less computations than an explicit feature mapping.

General scheme: compare objects by comparing substructures !

Application scenario:
each object is described by a set of features where the cardinality of the set can differ between objects.

Prominent examples:

- **computer vision:** extract features (image patches, gradients, histograms,...) at interesting points (variation of location and scale). Then the image is summarized by the set of extracted features.
- **natural language processing:** neglecting semantic information a text document simply consists of a set of words or sentences.

Kernels on sets II

Two approaches:

- directly compare two sets using a kernel defined on the components of the sets,
- count the number of occurrences of elements and compare the counts

bag-of-words representation

Kernels on sets III

Reminder: $2^{\mathcal{X}}$ is the powerset of $\mathcal{X}$, the set of all finite subsets of $\mathcal{X}$.

Proposition

Let $\mathcal{X}$ be a set and $k' : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive definite kernel on $\mathcal{X}$, then a kernel on finite subsets of $\mathcal{X}$, the set kernel, *$k : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$, is given by*

$$\forall\, A, B \in 2^{\mathcal{X}}, \qquad k(A,B) = \sum_{a \in A} \sum_{b \in B} k'(a,b).$$

Proof: Let $\Phi : \mathcal{X} \to \mathcal{H}_{k'}$ be the feature mapping associated to the kernel k'. Then using the linear mapping $\Phi_{2^{\mathcal{X}}} : 2^{\mathcal{X}} \to \mathcal{H}_{k'}$ defined as $A \to \Phi_{2^{\mathcal{X}}}(A) = \sum_{a \in A} k'(a, \cdot)$ we get

$$\begin{aligned}\langle \Phi_{2^{\mathcal{X}}}(A), \Phi_{2^{\mathcal{X}}}(B) \rangle_{\mathcal{H}_{k'}} &= \left\langle \sum_{a \in A} k'(a, \cdot), \sum_{b \in B} k'(b, \cdot) \right\rangle_{\mathcal{H}_{k'}} \\ &= \sum_{a \in A} \sum_{b \in B} \langle k'(a, \cdot), k'(b, \cdot) \rangle_{\mathcal{H}_{k'}} = \sum_{a \in A} \sum_{b \in B} k'(a,b) = k(A,B).\end{aligned}$$

The set kernel:

- adds up all similarities between elements of the sets.
- problems if cardinality varies very much $\Longrightarrow$ sets with large number of elements will be similar to every other set $\Longrightarrow$ normalization necessary,

$$\tilde{k}(A,B) := \frac{k(A,B)}{\sqrt{k(A,A)k(B,B)}} = \frac{\sum_{a\in A}\sum_{b\in B} k'(a,b)}{\sqrt{\sum_{a,a'\in A} k'(a,a')\sum_{b,b'\in B} k(b,b')}},$$

or

$$\tilde{k}(A,B) := \frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} k'(a,b),$$

- **Advantage:** two disjoint sets A and B $(A \cap B = \emptyset)$ can have a non-zero similarity value,
- the set kernel can be used for arbitrary sets not only subsets of $\mathcal{X}$.

Invariances via sets:

- classifier should be invariant under small transformations of the data (small rotations/translations in the case of handwritten digit recognition.
- add to each training object all its small transformations

$$\text{new object} = \text{old object} + \text{all transformations (set of objects)}$$

- apply set kernel to this set.

A simple set kernel not taking into account any structure of $\mathcal{X}$:

Proposition

Let $\mathcal{X}$ be some set. Then a kernel on finite subsets of $\mathcal{X}$, the intersection kernel*,*
$k : 2^{\mathcal{X}} \times 2^{\mathcal{X}} \to \mathbb{R}$*, is given by*

$$\forall\, A, B \in 2^{\mathcal{X}}, \qquad k(A, B) = |A \cap B|.$$

Proof: One can show that $\min\{x, y\}$ is a kernel on $\mathbb{R}_+$. For a finite set $\mathcal{X}$ one has

$$|A \cap B| = \sum_{x \in \mathcal{X}} \min\{A(x), B(x)\},$$

where $A(x)$ denotes the number of elements of type x in the set A. This finishes the proof since we add up valid kernels and the index set of the sum is **fixed**.

Kernels on sets VII

Taking into account both aspects ($M(\mathcal{X})$ denotes arbitrary sets consisting of elements in $\mathcal{X}$):

Proposition

Let $\mathcal{X}$ be a finite set and

- $k' : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *a positive definite kernel on* $\mathcal{X}$,
- $\overline{k} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ *a positive definite kernel on* $\mathbb{R}_+$.

Then the general set kernel *between arbitrary sets consisting of elements in* $\mathcal{X}$, $k : M(\mathcal{X}) \times M(\mathcal{X}) \to \mathbb{R}$, *is given by*

$$k(A, B) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{X}} k'(x, y)\overline{k}(A(x), B(y)),$$

where $A(x)$ is the number of times the element x is contained in set A.

Properties of the general set kernel:

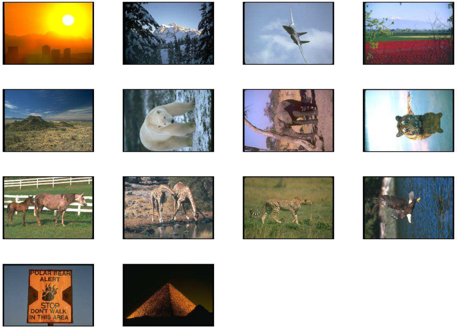
- comparison of arbitrary sets (the standard form is a histogram),
- integration of a complex weighting scheme depending on the similarity of the frequency of occurrence via $\overline{k}(A(x), B(y))$,
- integration of a given similarity measure on $\mathcal{X}$. This can be e.g. used to integrate semantic similarity when comparing texts.

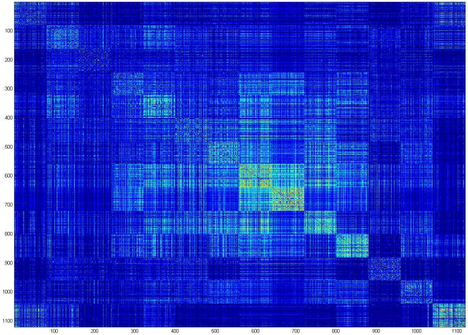
Normalization of the kernel or normalization of the counts $A(x)$ might be useful.

Problem:

- 14 categories of images (different animals, landscapes, airplanes, mountains),
- image representation: color histogram (set of colors !)
  (each channel in RGB is quantized into 16 levels - yielding a 4096 dimensional histogram).
- bag-of-colors representation.

Kernels on sets: Example II





- good block-diagonal structure of the kernel matrix,
- 10.4% error for a 14-class problem.