kernels (II)

presented by Virgil Pavlu

based on work by
Bernhard Scholkopf
Alexander Smola
Nello Cristianini
John Shaw-Taylor
Thorsten Joachims
where to read

Learning with Kernels
Support Vector Machines, Regularization, Optimization, and Beyond

Bernhard Schölkopf and Alexander J. Smola

An Introduction to Support Vector Machines and other kernel-based learning methods
this lecture

kernels
mercer conditions
SVM with kernels
designing kernels
feature extraction : kernel PCA
data similarities & dot product

- measurement of data similarities: a fundamental problem in ML

- reflects a priori knowledge of the problem/data

- dot product: a natural measure for similarity
  \[ \langle x \cdot y \rangle = \sum_i x_i \cdot y_i \]

- dot product amounts to being able to carry all geometric constructions formulated in terms of angles, lengths and distances

  \[ \cos(x, y) = \frac{\langle x \cdot y \rangle}{\|x\| \|y\|} \quad \|x\| = \sqrt{\langle x \cdot x \rangle} \]
feature space

- general measure for similarity
  \[ k : X \times X \to \mathbb{R}, \text{ symetric } k(x, y) = k(y, x) \]

- symmetry is too general, we want something that feels like dot product
  \[ \exists \Phi : X \to H \text{ mapping function} \]
  \[ k(x, y) = \Phi(x) \cdot \Phi(y) \]
  where \( H = \text{feature space (Hilbert space, supports dot product)} \)
\( \Phi \) extends the attribute space

- Input space
- Feature space
- \( a, b, c \)
- \( a, b, c, aa, ab, ac, bb, bc, cc \)
non-linear data separation

• i.e. when linear classifiers fail

• using a non-linear mapping $\Phi$ and a linear classifier in the feature space may succeed
feature space: example

input space: \( x = (x_1, x_2) \) (2 attributes)
feature space: \( \Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1) \) (6 attributes)
kernels

\[ \exists \Phi : X \rightarrow H, \ k : X \times X \rightarrow \mathbb{R} \]

\[ k(x, y) = k(y, x) \]
\[ k(x, y) = \Phi(x) \cdot \Phi(y) \]

\( H = \) feature space, \( \Phi = \) map(feature) function

- for which \( k \) there exits \( \Phi \) ?
- given \( k \), if \( \Phi \) exists, it may be not unique
this lecture

kernels

mercer conditions

SVM with kernels

designing kernels

feature extraction : kernel PCA
linear algebra

- \( \langle x \cdot Ay \rangle = \langle A^T x \cdot y \rangle \). A is **symmetric** if \( A = A^T \). then \( \langle x \cdot Ay \rangle = \langle Ax \cdot y \rangle \)

- A is **positive definite** if A is symmetric and satisfies
\[
\langle x \cdot Ax \rangle = x^T Ax = \sum_{i,j} x_i a_{ij} x_j \geq 0, \forall x
\]

- A is **unitary (orthogonal)** if \( A^T = A^{-1} \) or \( AA^T = I \). then
\[
\langle Ax \cdot Ay \rangle = \langle A^T Ax \cdot y \rangle = \langle A^{-1} Ax \cdot y \rangle = \langle x \cdot y \rangle
\]

- \( \det(A) \neq 0 \iff A \) has full rank \( \iff \exists A^{-1} \)
more linear algebra

• \( \lambda \) is eigenvalue of matrix \( A \) if there is a non-zero vector \( x \) (eigenvector) such that \( Ax = \lambda x \). then \( \det(A - \lambda I) = 0 \). eigenvectors are linear independent if eigenvalues are different

• \( \det(A) = \prod_i \lambda_i \). if a matrix is triangular/diagonal then its eigenvalues are exactly the diagonal entries

• if the eigenvectors \( V = (v_1^T, ..., v_n^T) \) are linear independent and form an orthonormal base and \( D = [\lambda_1, ..., \lambda_n] \) diagonal matrix then \( V^{-1}AV = V^T AV = D \Leftrightarrow A = VDV^T = VDV^{-1} \) (diagonalization). any symmetric matrix can be diagonalized

• SVD if \( A \) is \( m \times n \) then \( A = Q_1MQ_2^T \); \( Q_1, Q_2 \) orthogonal, \( M \) diagonal
kernel characterization
data dependent - $X$ finite

**Theorem** if the Gram matrix $K_{ij} = k(x_i, x_j)$ is positive definite then $k$ is a dot product: $\exists \Phi$ such that $k(x, y) = \Phi(x) \cdot \Phi(y)$

**Proof** $K$ positive definite $\Rightarrow K = SDS^T$ (diagonalization) where $S$ is orthogonal and $D$ is diagonal with non-negative entries then $k(x_i, x_j) = (SDS^T)_{ij} = \langle S_i \cdot DS_j \rangle = \langle \sqrt{DS_i} \cdot \sqrt{DS_j} \rangle$

take $\Phi(x_i) = \sqrt{DS_i}$
kernel characterization (converse)
data dependent - $X$ finite

**Theorem** if the kernel $k$ is a dot product $\exists \Phi$, $k(x, y) = \Phi(x) \cdot \Phi(y)$ then the Gram matrix $K_{ij} = k(x_i, x_j)$ is positive definite

**Proof** for any $\alpha \in \mathbb{R}^m$

$$\sum_{i,j=1}^{m} \alpha_i \alpha_j K_{ij} = \langle \sum_{i=1}^{m} \alpha_i \Phi(x_i), \sum_{j=1}^{m} \alpha_j \Phi(x_j) \rangle = \| \sum_{i=1}^{m} \alpha_i \Phi(x_i) \|^2 \geq 0$$

so $K$ is positive definite
mercer theorem

**Theorem [Mercer]** Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^n$. Suppose $\mathcal{K}$ is a continuous symmetric function such that

$$\int_{\mathcal{X}} \int_{\mathcal{X}} \mathcal{K}(x, z) f(x) f(z) dx dz \geq 0$$

for all $f \in L_2(\mathcal{X})$. Then, $\mathcal{K}(x, z)$ can be expanded in a uniformly convergent series

$$\mathcal{K}(x, z) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(z)$$

in terms of the eigenfunctions $\phi_j \in L_2(\mathcal{X})$ of $(T_{\mathcal{K}}f)(\cdot) = \int_{\mathcal{X}} \mathcal{K}(\cdot, x) f(x) dx$ normalized so that $||\phi_j||_{L_2} = 1$ and positive associated eigenvalues $\lambda_j \geq 0$. 

valid kernels

- $\mathcal{K}(x, z) = \mathcal{K}_1(x, z) + \mathcal{K}_2(x, z)$
- $\mathcal{K}(x, z) = a\mathcal{K}_1(x, z)$
- $\mathcal{K}(x, z) = \mathcal{K}_1(x, z)\mathcal{K}_2(x, z)$
- $\mathcal{K}(x, z) = f(x)f(z)$
- $\mathcal{K}(x, z) = \mathcal{K}_3(\phi(x), \phi(z))$

$p(x)$ a polynomial with positive coefficients

- $\mathcal{K}(x, z) = p(\mathcal{K}_1(x, z))$
- $\mathcal{K}(x, z) = \exp(\mathcal{K}_1(x, z))$
- $\mathcal{K}(x, z) = \exp(-\|x - z\|^2/\sigma^2)$
dot product kernels

\[ k(x, y) = k(\langle x, y \rangle) \]

**Theorem**

- The function \( k \) of the dot product kernel must satisfy
  \[ k(t) \geq 0, k'(t) \geq 0 \text{ and } k'(t) + tk''(t) \geq 0 \forall t \geq 0 \]
  in order to be a positive definite kernel. That may still be insufficient.

- If \( k \) is a power series expansion

  \[
  k(t) = \sum_{n=0}^{\infty} a_n t^n
  \]

  then \( k \) is a positive definite kernel iff \( \forall n, a_n \geq 0 \)
this lecture

kernels
mercer conditions
**SVM with kernels**
designing kernels
feature extraction : kernel PCA
SVMs

\[ \{ x \mid \langle w, x \rangle + b = -1 \} \]

\[ \{ x \mid \langle w, x \rangle + b = +1 \} \]

\[ y_i = -1 \]

\[ y_i = +1 \]

\[ x_1 \]

\[ x_2 \]

\[ w \]

\[ \langle w, x_1 \rangle + b = +1 \]

\[ \langle w, x_2 \rangle + b = -1 \]

\[ \Rightarrow \quad \langle w, (x_1 - x_2) \rangle = 2 \]

\[ \Rightarrow \quad \langle \frac{w}{\|w\|}, (x_1 - x_2) \rangle = \frac{2}{\|w\|} \]
plug a kernel into SVM after a long discussion on optimization theory...

the primal problem
minimize $\langle w \cdot w \rangle$
subject to $y_i(\langle w \cdot x_i \rangle + b) \geq 1, \forall i$

the dual problem
maximize $P(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle$
subject to $\sum_{i=1}^{m} y_i \alpha_i = 0, \alpha_i \geq 0, \forall i$

kernel trick replace the dot product $\langle x_i \cdot x_j \rangle$ with a kernel $k(x_i, x_j)$:
maximize

$$P(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j k(x_i, x_j)$$

subject to $\sum_{i=1}^{m} y_i \alpha_i = 0, \alpha_i \geq 0, \forall i$
SVM with kernels

\[ \sigma \left( \sum \right) \]

weights

\( \langle \Phi(x), \Phi(x_i) \rangle = k(x, x_i) \)

mapped vectors \( \Phi(x_i), \Phi(x) \)

support vectors \( x_1 \ldots x_n \)

test vector \( x \)
the kernel trick

maximize

\[ P(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \]

subject to \( \sum_{i=1}^{m} y_i \alpha_i = 0, \alpha_i \geq 0, \forall i \)

• we need only the kernel \( k \), not \( \Phi \) - that's good...
• any algorithm that only depends on dot products (rotationally invariant) can be kernelized
• any algorithm that is formulated in terms of positive definite kernel(s) supports a kernel-replace
• math was around for long time (1940s Kolgomorov, Aronszajn, Schoenberg) but the practical importance was underestimated
SVM, concept class, good kernels

$C$ a concept class $=$ set of concepts

a kernel is **complete** if it is "fine-grained" enough

$k(x_i, \cdot) = k(x_j, \cdot) \Rightarrow c(x_i) = c(x_j), \forall c \in C$

a kernel is **correct(linear-good) wrt to $C$** if an SVM with perfect separation can be learned with it

$\forall c \in C, \exists w$ such that $\sum_i w_i k(x_i, x) \geq 0 \Leftrightarrow c(x)$
this lecture

kernels
mercer conditions
SVM with kernels
designing kernels
feature extraction : kernel PCA
polynomial kernel

**Theorem** define the map $\mathbf{x} \rightarrow C_d(\mathbf{x})$ where $C_d(\mathbf{x})$ the vector consisting in all possible $d^{th}$ degree ordered products of the entries of $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ then $\langle C_d(\mathbf{x}), C_d(\mathbf{y}) \rangle = \langle \mathbf{x}, \mathbf{y} \rangle^d$

$$k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$$

polynomial kernel

- invariant to group of all orthogonal transformations (rotations, mirroring)
polynomial kernel: toy example

use the map \( \mathbf{x} = (x_1, x_2) \rightarrow \Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \)
ellipse from 2D-input space becomes hyperplane into 3D-feature space

note \( C_2(\mathbf{x}) = (x_1^2, x_2^2, x_1x_2, x_2x_1) \) maps data in a 4D-feature space
but it generates the same kernel
\[
k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = \langle C_2(\mathbf{x}), C_2(\mathbf{y}) \rangle = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1x_2y_2
\]
Gaussian Radial Basis Function kernel

\[ k(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right) \]

more general \( k(x, y) = f(d(x, y)) \)

where \( d \) is a metric on \( X \) and \( f \) is a function on \( \mathbb{R}^+_0 \); usually \( d \) arises from dot product \( d(x, y) = ||x - y|| \)

- invariant on translations \( k(x, y) = k(x + z, y + z) \)
- \( \cos(\angle(\Phi(x), \Phi(y))) = \langle \Phi(x), \Phi(y) \rangle = k(x, y) \geq 0 \Rightarrow \) enclosed angle between any 2 mapped points is smaller than \( \pi/2 \)

**Theorem** if \( X = \{x_1, x_2, ..., x_m\} \) all distinct and \( \sigma > 0 \) then the matrix \( K_{ij} = \exp(-\frac{||x_i - x_j||^2}{2\sigma^2}) \) has full rank \( \Rightarrow \) \( \Phi(x_1), \Phi(x_2), ..., \Phi(x_m) \) are linear independent
RBF kernel SVM
<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>KFD</th>
<th>RBF</th>
<th>AB</th>
<th>$\text{AB}_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>11.5±0.07</td>
<td>10.8±0.05</td>
<td>10.8±0.06</td>
<td>12.3±0.07</td>
<td>10.9±0.04</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>26.0±0.47</td>
<td>25.8±0.46</td>
<td>27.6±0.47</td>
<td>30.4±0.47</td>
<td>26.5±0.45</td>
</tr>
<tr>
<td>Diabetes</td>
<td>23.5±0.17</td>
<td>23.2±0.16</td>
<td>24.3±0.19</td>
<td>26.5±0.23</td>
<td>23.8±0.18</td>
</tr>
<tr>
<td>German</td>
<td>23.6±0.21</td>
<td>23.7±0.22</td>
<td>24.7±0.24</td>
<td>27.5±0.25</td>
<td>24.3±0.21</td>
</tr>
<tr>
<td>Heart</td>
<td>16.0±0.33</td>
<td>16.1±0.34</td>
<td>17.6±0.33</td>
<td>20.3±0.34</td>
<td>16.5±0.35</td>
</tr>
<tr>
<td>Image</td>
<td>3.0±0.06</td>
<td>3.3±0.06</td>
<td>3.3±0.06</td>
<td>2.7±0.07</td>
<td>2.7±0.06</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.7±0.01</td>
<td>1.5±0.01</td>
<td>1.7±0.02</td>
<td>1.9±0.03</td>
<td>1.6±0.01</td>
</tr>
<tr>
<td>F. Sonar</td>
<td>32.4±0.18</td>
<td>33.2±0.17</td>
<td>34.4±0.20</td>
<td>35.7±0.18</td>
<td>34.2±0.22</td>
</tr>
<tr>
<td>Splice</td>
<td>10.9±0.07</td>
<td>10.5±0.06</td>
<td>10.0±0.10</td>
<td>10.1±0.05</td>
<td>9.5±0.07</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.8±0.22</td>
<td>4.2±0.21</td>
<td>4.5±0.21</td>
<td>4.4±0.22</td>
<td>4.6±0.22</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.4±0.10</td>
<td>23.2±0.20</td>
<td>23.3±0.13</td>
<td>22.6±0.12</td>
<td>22.6±0.12</td>
</tr>
<tr>
<td>Twonorm</td>
<td>3.0±0.02</td>
<td>2.6±0.02</td>
<td>2.9±0.03</td>
<td>3.0±0.03</td>
<td>2.7±0.02</td>
</tr>
<tr>
<td>Waveform</td>
<td>9.9±0.04</td>
<td>9.9±0.04</td>
<td>10.7±0.11</td>
<td>10.8±0.06</td>
<td>9.8±0.08</td>
</tr>
</tbody>
</table>
Fisher kernel

- Knowledge about objects in form of a generative probability model
- Deals with missing/incomplete data, uncertainty, variable length

**Family of generative models** (density functions)
$p(x|\theta)$, smoothly parametrized by $\theta = (\theta^1, ..., \theta^r)$; $l(x, \theta) = \ln p(x|\theta)$

**Score** $V_\theta(x) := (\delta_{\theta^1}l(x, \theta), ..., \delta_{\theta^r}l(x, \theta)) = \nabla_\theta l(x, \theta) = \nabla_\theta \ln p(x|\theta)$

**Fisher information matrix** $I := \mathbb{E}_p[V_\theta(x)V_\theta(x)^T]$
$I_{ij} = \mathbb{E}_p[\delta_{\theta^i} \ln p(x|\theta) \cdot \delta_{\theta^j} \ln p(x|\theta)]$, $\mathbb{E}_p$ is called **Fisher information metric**

**Fisher kernel**
$K_I(x, y) := V_\theta(x)^T I^{-1} V_\theta(y)$

**Natural kernel** $M$ positive definite matrix
$K_M^{nat}(x, y) := V_\theta(x)^T M^{-1} V_\theta(y)$
[information] diffusion kernel
- local relationships

the exponential of a squared matrix $H$ is
$$e^{\beta H} = \lim_{n \to \infty} \left( 1 + \frac{\beta H}{n} \right)^n = I + \beta H + \frac{\beta^2}{2!} H^2 + \frac{\beta^3}{3!} H^3 + ...$$

**exponential kernel** $K_{\beta} = e^{\beta H}$, $\frac{\delta K_{\beta}}{\delta \beta} = HK_{\beta}$ (heat eq)

**diffusion kernel** on graph: consider

$H_{ij} = 1$ if $i, j$; $-d_i$ (degree) if $i = j$; 0 otherwise

$w^T H w = -\sum_{i, j \in E} (w_i - w_j)^2$ negative semidefinite

$-H = $Laplacian of the graph
two approaches to kernel design

**model driven** - encodes knowledge about domain
- polynomial, Gaussian
- from generative models: Fisher kernel
- local relationships: diffusion kernel

**syntax driven** - exploits structure of the problem
- terms: convolution kernel
- text classification: string kernel
- tree kernel
- particularly useful for non-vectorial data
convolution kernel
kernel between composite objects building on similarities of resp. parts
\[ k_d : X_d \times X_d \to \mathbb{R}, \text{ } R\text{-relation. define the } R\text{-convolution kernel} \]

\[ (k_1 \ast k_2 \ast \ldots \ast k_D)(x, y) := \sum_{R} \prod_{d=1}^{D} k_d(x_d, y_d) \]

where the sum runs over all possible decompositions of \( x \to (x_1, x_2, \ldots, x_D) \)
and of \( y \to (y_1, y_2, \ldots, y_D) \) s.t. \( R(x, x_1, x_2, \ldots, x_D) \) and \( R(y, y_1, y_2, \ldots, y_D) \)
• proved valid if \( R \) is finite

ANOVA kernel (analysis of variance)
if \( X = S^N \) and \( k^{(i)} \) kernel on \( S \times S \) for \( i = 1, 2, \ldots, N \), the ANOVA kernel
of order \( D \) is

\[ k_D(x, y) := \sum_{1\leq i_1<\ldots<i_D\leq N} \prod_{d=1}^{D} k_{i_d}^{d}(x_{i_d}, y_{i_d}) \]
string kernel - similarities between two documents

\[ \sum = \text{alphabet}, \quad \sum^n = \text{set of all strings of length } n \]

for a given index sequence \( \mathbf{i} = (1 \leq i_1 < i_2 < \ldots < i_r \leq |s|) \)

define \( s(\mathbf{i}) := s(i_1)s(i_2)\ldots s(i_r) \) and \( l_\mathbf{s}(\mathbf{i}) = i_r - i_1 + 1 \geq r \)

example \( s = \text{fast food}, \mathbf{i} = (2, 3, 9) \Rightarrow s(\mathbf{i}) = \text{asd}, l_\mathbf{s}(\mathbf{i}) = 9 - 2 + 1 = 8 \)

\[ 0 < \lambda \leq 1 \text{ parameter, define } [\Phi_n(s)] \text{ a map with } |\sum^n| \text{ components} \]

\[ [\Phi_n(s)]_u = \sum_{\mathbf{i} : s(\mathbf{i}) = u} \lambda^{l_\mathbf{s}(\mathbf{i})} \]

example \( [\Phi_3(\text{Nasdaq})]_{\text{asd}} = \lambda^3, \quad [\Phi_3(\text{lass das})]_{\text{asd}} = 2\lambda^5 \)

the kernel induced

\[ k_n(s, t) = \sum_{u \in \sum^n} [\Phi_n(s)]_u [\Phi_n(t)]_u = \sum_{u \in \sum^n} \sum_{(i, j) : s(i) = t(j) = u} \lambda^{l_\mathbf{s}(i)} \lambda^{l_\mathbf{t}(j)} \]

\[ k := \sum_n c_n k_n \text{ linear combination of kernels on different substring-lengths} \]
tree kernel

- encode a tree as a string by traversing in preorder and parenthesing
- substrings correspond to subset trees
- tag can be computed in loglinear time
- then use a string kernel

$$\text{tag}(T) = (A(B(C)(D)))(E)$$
kernels correspond to

- similarity measure for the data
- linear representation of the data
- function space for learning
- covariance function for correlated observations
- prior over the set of functions

the kernel is the prior knowledge we have about the problem and its solution - no free lunch here
this lecture

kernels
mercer conditions
SVM with kernels
designing kernels

feature extraction : kernel PCA
Principal Component Analysis
technique for extracting structure from possible high-dim data sets
given observations $x_i \in \mathbb{R}^N, i = 1, \ldots, m$
centered: $\sum_{i=1}^{m} x_i = 0$
form the covariance matrix $C = \frac{1}{m} \sum_{j=1}^{m} x_j x_j^T$, positive definite

$C$ can be diagonalized with non negative eigenvalues. To do this, solve
the eigenvalue eq $\lambda v = Cv$ for $\lambda \geq 0$ and non-zero eigenvectors $v \in \mathbb{R}^N$
equation becomes

$$\lambda v = Cv = \frac{1}{m} \sum_{j=1}^{m} \langle x_j, v \rangle x_j$$

all $v$ with $\lambda \neq 0$ lie in the span of $x_1, \ldots, x_m$ hence the eigenvalue eq
becomes $\lambda \langle x_i, v \rangle = \langle x_i, Cv \rangle, \forall i$
kernel PCA

$\Phi : X \rightarrow H$ (possibly nonlinear) map, cenetred $\sum_{i=1}^{m} \Phi(x_i) = 0$.

the covariance matrix $C = \frac{1}{m} \sum_{j=1}^{m} \Phi(x_j)\Phi(x_j)^T$.

as in PCA, we need to find the eigenvalues and eigenvectors satisfying $\lambda v = Cv$. note that solutions lie in the spam of $\Phi(x_1), \ldots, \Phi(x_m)$ or $v = \sum_{i=1}^{m} \alpha_1 \Phi(x_i)$ and equation is equiv to $\lambda \langle \Phi(x_i), v \rangle = \langle \Phi(x_i), Cv \rangle, \forall i$ which becomes

$$\lambda \sum_{i=1}^{m} \alpha_i \langle \phi(x_n), \Phi(x_i) \rangle = \frac{1}{m} \sum_{i=1}^{m} \alpha_i \langle \Phi(x_n), \sum_{j=1}^{m} \Phi(x_j) \langle \Phi(x_j), \Phi(x_i) \rangle \rangle$$

if $K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle$ (Gram matrix) then we need to find non-zero solutions of $m\lambda K\alpha = K^2\alpha$ which are between solutions of $m\lambda \alpha = K\alpha$
kernel PCA - properties

kernel PCA is the orthogonal basis transformation in $\mathbf{H}$ with following properties (assuming eigenvectors in descending order of eigenvalues):

- first $q$ principal components (proj. on eigenvectors) carry more variance that any other $q$ orthogonal directions
- the mean-squared approx. error when representing $(x_i)$ by the $q$ first principal components is minimal
- the principal components are uncorrelated
- the first $q$ principal components have max mutual information
- connection with SVM : the $n^{th}$ KPCA feature extractor, scaled by $1/\lambda_n$ is optimal among all feature extractions, in the sense that it has minimal weight vector norm in the RKHS $\mathbf{H}$, $\|v\|^2 = \sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j)$ subject to orthogonality and unit variance set of outputs when applied to training set $(x_i)$
important things not covered

- regularization
- kernel fisher discriminant
- bayesian kernel methods
- locality-improved kernels
END

don’t look after this slide
bayesian kernel methods
kernels and Gaussian processes
kernel fisher discriminant