Advanced Topics in Machine Learning (600.692)
Homework 7: Kernel PCA and Manifold Learning

Instructor: René Vidal

Due Date: 04/11/2014, 11.59PM Eastern

READING MATERIAL: Chapters 4 and 5 of GPCA book.

1. (a) Let $k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{y})$ for some embedding function $\phi : \mathbb{R}^D \to \mathbb{R}^M$. Show that k is a symmetric positive semi-definite kernel.

   (b) Consider the polynomial kernel in $[-1, 1]^2 \times [-1, 1]^2$ defined as $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y})^2 = (x_1 y_1 + x_2 y_2)^2$. Define the operator

$$\mathcal{L}(f)(\boldsymbol{x}) = \int k(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y}) d\boldsymbol{y}. \tag{1}$$

   Show that the eigenfunctions of $\mathcal{L}$ corresponding to nonzero eigenvalues are of the form $\psi(\boldsymbol{x}) = c_1 x_1^2 + c_2 x_1 x_2 + c_3 x_2^2$. Show that there are three such eigenfuctions, where (c_1, c_2, c_3) and λ are obtained from

$$\begin{bmatrix} 4/5 & 0 & 4/9 \\ 0 & 8/9 & 0 \\ 4/9 & 0 & 4/5 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \lambda \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}. \tag{2}$$

2. Implement the KPCA algorithm for an arbitrary kernel function `kernel.m`. The format of your function should be as follows.

Function	**`[y]=kpca(x,d,kernel,params)`**
Parameters	
`x`	$D \times N$ matrix whose columns are the data points
`d`	dimension of the projected dataset
`kernel`	name of the MATLAB function that computes the kernel `k = kernel(x1,x2,params)`
`params`	parameters needed by the kernel function, such as the degree in the polynomial kernel or the standard deviation in the Gaussian kernel
Returned values	
`y`	$d \times N$ matrix containing the projected coordinates
Description	
Computes the kernel principal components of a set of points.	

   Also implement the functions `k = poly_kernel(x1,x2,n)` for the polynomial kernel $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^\top \boldsymbol{x}_2)^n$ and `k = gauss_kernel(x1,x2,sigma)` for the Gaussian kernel $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/\sigma^2)$, where $k \in \mathbb{R}^{N \times N}$ and $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{D \times N}$. Try your code on the example given in class. The code generating the data can be found at http://www.kernel-machines.org/code/kpca_toy.m

3. **Properties of the Veronese map.**

   Consider the Veronese map $\nu_n : [x_1, \ldots, x_D]^T \mapsto [\ldots, \boldsymbol{x}^n, \ldots]^T$ where $\boldsymbol{x}^n = x_1^{n_1} x_2^{n_2} \ldots x_D^{n_D}$ ranges over all monomials of degree $n = \sum_{i=1}^{D} n_i$ in the variables $x_1, x_2, \ldots, x_D$, sorted in the degree-lexicographic order, and let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$.

   (a) **Inner product invariance:** Show that the polynomial kernel $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{y}^\top \boldsymbol{x})^n$ can be written in terms of the Veronese map as $k(\boldsymbol{x}, \boldsymbol{y}) = \nu_n(\boldsymbol{y})^\top M \nu_n(\boldsymbol{x})$, where $M \in \mathbb{R}^{M_n(D) \times M_n(D)}$ is a diagonal matrix, and its $(n_1, n_2, \ldots, n_D)$th entry is $\frac{n!}{n_1! n_2! \ldots n_D!}$ with $\sum_{i=1}^{D} n_i = n$.

Hint: Use the Multinomial Theorem.

(b) **Linear invariance:**

i. Show that $\nu_n(\alpha\boldsymbol{x}+\boldsymbol{y}) = \sum_{i=0}^{n} \alpha^i f_i(\boldsymbol{x},\boldsymbol{y})$ where $f_i(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}^{M_n(D)}$ is a bi-homogenous polynomial of degree i in $\boldsymbol{x}$ and $(n-i)$ in $\boldsymbol{y}$ for $i = 0,\dots,n$.

ii. Let S_n be the space of homogeneous polynomials of degree n in D variables. Define the transformation $T : S_n \to S_n$, such that $T(p_n(\boldsymbol{x})) = p_n(A\boldsymbol{x})$, where $A \in \mathbb{R}^{D\times D}$. Show that the transformation T is linear.

iii. Show that for all $A \in \mathbb{R}^{D\times D}$ there exists an $\tilde{A} \in \mathbb{R}^{M_n(D)\times M_n(D)}$ such that for all $\boldsymbol{x}, \nu_n(A\boldsymbol{x}) = \tilde{A}\nu_n(\boldsymbol{x})$.

(c) **Rotation invariance:** Show that for $D = 3$ and all $R \in SO(3)$, there exists $\tilde{R} \in SO(M_n(D))$ such that for all $\boldsymbol{x}, \nu_n(R\boldsymbol{x}) = \tilde{R}\nu_n(\boldsymbol{x})$.

4. **(Multiple Algebraic Subspace Clustering).** For each $f = 1,2,\dots,F$, let $\{\boldsymbol{x}_{fj} \in \mathbb{R}^D\}_{j=1}^N$ be a collection of N points lying in n hyperplanes with normal vectors $\{\boldsymbol{b}_{fi}\}_{i=1}^n$. Assume that for each $j = 1,2,\dots,N$, the F points $\boldsymbol{x}_{1j},\boldsymbol{x}_{2j},\dots,\boldsymbol{x}_{Fj}$ correspond to each other. That is, for each $j = 1,2,\dots,N$ there is an $i = 1,2,\dots,n$ such that for all $f = 1,2,\dots,F$, we have $\boldsymbol{b}_{fi}^\top \boldsymbol{x}_{fj} = 0$. Propose an extension of the ASC algorithm that computes the normal vectors in such a way that $\boldsymbol{b}_{1i},\boldsymbol{b}_{2i},\dots\boldsymbol{b}_{Fi}$ correspond to each other.

5. Let $\{\boldsymbol{x}_j \in \mathbb{R}^D\}_{j=1}^N$ be a collection of points lying in n affine subspaces

$$S_i = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{\mu}^i + U_{d_i}^i \boldsymbol{y}\} \qquad i = 1,\dots,n$$

of dimensions d_i, where $\boldsymbol{\mu}^i \in \mathbb{R}^D$, $U_{d_i}^i \in \mathbb{R}^{D\times d_i}$ has orthonormal columns, and $\boldsymbol{y} \in \mathbb{R}^{d_i}$. Assume that within each subspace S_i the data is distributed around m_i cluster centers $\{\mu_{ik} \in \mathbb{R}^D\}_{i=1\dots n}^{k=1\dots m_i}$.

(a) Assume that n, d_i and m_i are known and propose a clustering algorithm similar to K-means and K-subspaces to estimate the model parameters $\boldsymbol{\mu}^i$, $U_{d_i}^i$, $\boldsymbol{y}_j^i$ and μ_{ik}, and the segmentation of the data according to the $\sum_{i=1}^n m_i$ groups. More specifically, write down the cost function to be minimized, the constraints among the model parameters (if any), and use Lagrange optimization to find the optimal model parameters given the segmentation.

(b) Assume that n, d_i and m_i are unknown. How would you modify the cost function of part a)?

Submission instructions. Please follow the same instructions as in HW1.