HW 4: Advanced Topics in Machine Learning

Instructor: René Vidal, E-mail: rvidal@cis.jhu.edu

Due 5/7/10 in class

1. **Affine motion segmentation.**

   (a) Use the function `gpca` from Homework 3 to segment the point correspondences of the following five video sequences in the course webpage: i) Kanatani1, ii) Kanatani2, iii) Kanatani3, iv) three-cars, v) can-book. You can use the provided function `loadSequence.m` to load the sequences and form the matrix of trajectories. Recall that you will need to project the data in $\mathbb{R}^{2F}$ onto a subspace of dimension d. What is the value for d? Assume the number of groups is known. Plot the three principal components of the data with different colors for the different groups. Report the percentage of misclassified point trajectories.

   (b) Repeat part (b) using the function `ksubspaces` that you implemented in Homework 4. Use the result of GPCA from part (b) to initialize K-subspaces. Use both the data without projection, i.e., the data in $\mathbb{R}^{2F}$, and the projected data in $\mathbb{R}^d$ as the input to K-subspaces. Which one is better, projecting or not, and why?

2. **Face Clustering with varying illumination:** A simple model for the images of n Lambertian faces taken under several illumination conditions is that they live in n 3-dimensional subspaces of $\mathbb{R}^P$, where P is the number of pixels. It follows that clustering a set of images of multiple faces according to which individuals the image belongs to is a subspace clustering problem. Load the set of images given in the course web-page. You will need to use the provided function `loadImage.m` to load all the images of individuals from 1 to 3 under the illumination conditions 1 to 8. Next, reduce the dimension using PCA to the first four principal components. Now, assume the number of groups is known, i.e. $n = 3$, and segment the faces using the functions `gpca` and `ksubspaces` that you implemented in Homework 3. Use also `ksubspaces` initialized by GPCA. Plot the first three components of the low-dimensional representation and report the percentage of incorrectly classified images.

3. **SVD-based subspace clustering**: Let $\{\boldsymbol{x}_j \in \mathbb{R}^D\}_{j=1}^N$ be a set of points drawn from a union of n independent subspaces $\{S_i\}_{i=1}^n$, i.e.,
$$r = \dim(\cup_{i=1}^n S_i) = \sum_{i=1}^n \dim(S_i) = \sum_{i=1}^n d_i. \tag{1}$$
Let $U_i \in \mathbb{R}^{D\times d_i}$ be a basis for subspace S_i, so that the N_i points in subspace S_i can be written as $X_i = U_i Y_i$, where $Y_i \in \mathbb{R}^{d_i \times N_i}$ is the low-dimensional representation of the data points in subspace S_i. Show that the data matrix $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$ can be factorized as
$$X = \begin{bmatrix} U_1, U_2, \cdots, U_n \end{bmatrix} \begin{bmatrix} Y_1 & 0 & \cdots & 0 \\ 0 & Y_2 & & \\ & & \ddots & 0 \\ 0 & \cdots & 0 & Y_n \end{bmatrix} \Gamma. \tag{2}$$
where $\Gamma \in \mathbb{R}^{N\times N}$ is a permutation matrix sorting the data points according to the subspaces they belong to, i.e., $X = [X_1, X_2, \cdots X_n]\Gamma$. Let $X = U\Sigma V^\top$ be the rank r SVD of the data matrix X, where $V \in \mathbb{R}^{r\times N}$. Let $Q = VV^\top \in \mathbb{R}^{N\times N}$. Show that $Q_{ij} = 0$ if points i and j are in different subspaces. Suggest an algorithm for clustering independent subspaces based on the SVD of the data matrix.

4. **GPCA for 2 hyperplanes**: Consider the problem of segmenting n data points $\{\boldsymbol{x}_j\}_{j=1}^N$ lying in two hyperplanes in $\mathbb{R}^D$ with normal vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$. Show that the data points satisfy the equation $\boldsymbol{x}^\top B \boldsymbol{x} = 0$, where $B = \boldsymbol{b}_1\boldsymbol{b}_2^\top + \boldsymbol{b}_2\boldsymbol{b}_1^\top \in \mathbb{R}^{D\times D}$. Write down a linear system for estimating B from data points. Show that if $\boldsymbol{b}_1 \neq \alpha \boldsymbol{b}_2$, for any $\alpha \neq 0$, then B has two nonzero eigenvalues λ_1 and λ_2 and $D-2$ zero eigenvalues. Show that $\lambda_1\lambda_2 < 0$. Show that one can estimate the normal vectors from B as
$$\begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \sqrt{|\lambda_1|} & \text{sign}(\lambda_1)\sqrt{|\lambda_1|} \\ \sqrt{|\lambda_2|} & \text{sign}(\lambda_2)\sqrt{|\lambda_2|} \end{bmatrix} \tag{3}$$
where $U_1 \in \mathbb{R}^D$ and $U_2 \in \mathbb{R}^D$ are the eigenvectors of B corresponding to the nonzero eigenvalues λ_1 and λ_2.

5. **Clustering linear from bilinear varieties**: Let $S = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^D \times \mathbb{R}^D : \boldsymbol{u}^\top \boldsymbol{x} = 0 \vee \boldsymbol{y}^\top A \boldsymbol{x} = 0\}$, where $\boldsymbol{u} \neq \mathbf{0}$ and $e_D^\top A = e_D^\top = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$.

   (a) Find a polynomial $p(\boldsymbol{x}, \boldsymbol{y})$ that vanishes on S. How many independent monomials are there in p?

   (b) Show that p can be written as $p(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}^\top \mathcal{M} \nu_2(\boldsymbol{x})$. How is $\mathcal{M}$ related to A and $\boldsymbol{u}$?

   (c) If $D = 3$, write the $\mathcal{M}$ explicitly, and show how one can compute $\boldsymbol{u}$ and A from the entries of $\mathcal{M}$.

   (d) Let $\boldsymbol{X} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \in S\}_{i=1}^N$ be a given data set. Derive an algorithm to compute $\mathcal{M}$ from $\boldsymbol{X}$. Then, derive an algorithm to compute A and $\boldsymbol{u}$ from the derivatives of p with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$.
   Hint: *Explicitly write down the derivatives of the polynomial $p(\boldsymbol{x}, \boldsymbol{y})$ and inspect the values of these derivatives when $\boldsymbol{u}^\top \boldsymbol{x} = 0$ and when $\boldsymbol{y}^\top A \boldsymbol{x} = 0$. Also, making clever canonical choices for $\boldsymbol{x}$ and $\boldsymbol{y}$ can make the solution easier.*

6. **Kernel GPCA**: Recall that Kernel PCA allows one to compute the principal components of the embedded data matrix $\Phi = [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \cdots, \phi(\boldsymbol{x}_N)]$ from the kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{y})$. In the case of the polynomial embedding, this reduces the calculations of the SVD of the covariance matrix, which is $M_n(D) \times M_n(D)$, to the SVD of the kernel matrix, which is $N \times N$. This is much more economic when D is large so that $M_n(D) \gg N$. This observation is usually referred to as the *kernel trick*.

   Recall also that the GPCA algorithm finds polynomials of the form $p_n(\boldsymbol{x}) = \boldsymbol{c}^\top \nu_n(\boldsymbol{x})$, whose coefficients $\boldsymbol{c}$ are in the null space of the embedded data matrix $\Phi = [\nu_n(\boldsymbol{x}_1), \nu_n(\boldsymbol{x}_2), \cdots, \nu_n(\boldsymbol{x}_N)]$.

   Recall now that the scaled Veronese map induces the polynomial kernel $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\top \boldsymbol{y})^n = \nu_n(\boldsymbol{x})^\top \nu_n(\boldsymbol{y})$.

   These three facts suggest that one may be able to solve the GPCA problem using the kernel trick. Specifically, find a way of computing a basis for each of the subspaces from the kernel matrix

$$K = \begin{bmatrix} (\boldsymbol{x}_1^\top \boldsymbol{x}_1)^n & (\boldsymbol{x}_1^\top \boldsymbol{x}_2)^n & \cdots & (\boldsymbol{x}_1^\top \boldsymbol{x}_N)^n \\ (\boldsymbol{x}_2^\top \boldsymbol{x}_1)^n & (\boldsymbol{x}_2^\top \boldsymbol{x}_2)^n & \cdots & (\boldsymbol{x}_2^\top \boldsymbol{x}_N)^n \\ \vdots & & & \vdots \\ (\boldsymbol{x}_N^\top \boldsymbol{x}_1)^n & (\boldsymbol{x}_N^\top \boldsymbol{x}_2)^n & \cdots & (\boldsymbol{x}_N^\top \boldsymbol{x}_N)^n \end{bmatrix} \tag{4}$$

   without having to explicitly compute vectors in $M_n(D)$. You may assume that the subspaces are independent. You may also assume $n = 2$ if you like.