HW 3: Advanced Topics in Machine Learning

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

Due 3/31/10 in class

1. **Robust PCA.** Consider the problem of decomposing a given matrix X into the sum of a low rank matrix L and a sparse matrix E by solving the following optimization problem

$$\min \ \|L\|_* + \lambda \|E\|_1 \tag{1}$$
$$\text{s.t. } X = L + E \tag{2}$$

The augmented Lagrangian method for solving this problem is given by

$$\max_{\Lambda} \min_{L,E} \quad \|L\|_* + \lambda \|E\|_1 + \text{trace}(\Lambda^\top (X - L - E)) + \frac{\mu}{2} \|X - L - E\|_F^2. \tag{3}$$

where Λ is a matrix of Lagrange multipliers and $\mu > 0$ is a fixed user-specified parameter. Show that this optimization problem can be solved using the following (coordinate ascent/descent) iterative procedure

- Initialize $E_0 = \Lambda_0 = 0$.
- While not converged do
  - $L_{k+1} = \mathcal{D}_{\mu^{-1}}(X - E_k - \mu^{-1}\Lambda_k)$
  - $E_{k+1} = \mathcal{S}_{\lambda\mu^{-1}}(X - L_{k+1} + \mu^{-1}\Lambda_k)$
  - $\Lambda_{k+1} = \Lambda_k + \mu(X - L_{k+1} - E_{k+1})$
- end while

Here $\mathcal{S}_\tau(x) = \text{sign}(x)\max(|x| - \tau, 0)$ is the shrinkage operator (which extends to matrices by applying it to each entry) and $\mathcal{D}_\tau(X)$ is the singular value thresholding operator given by $\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^\top$, where $X = U\Sigma V^\top$ is the SVD of X.

2. **Properties of the Veronese map.** Consider the Veronese map $\nu_n : [x_1, \ldots, x_D]^\top \mapsto [\ldots, \boldsymbol{x}^n, \ldots]^\top$ 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) **Number of monomials:** Show that the number of linearly independent monomials is given by

$$M_n(D) = \binom{n + D - 1}{n}$$

   (b) **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.

   (c) **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 $i = 0, \ldots, n$.

      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 $\boldsymbol{x}, \nu_n(A\boldsymbol{x}) = \tilde{A}\nu_n(\boldsymbol{x})$.

   (d) **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 $\boldsymbol{x}, \nu_n(R\boldsymbol{x}) = \tilde{R}\nu_n(\boldsymbol{x})$.

3. **Joint Central and Subspace Clustering.** Let $\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^P$ be a collection of points lying in n affine subspaces

$$S_j = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{x}_0^j + U_{d_j}^j \boldsymbol{y}\} \qquad j = 1, \ldots, n$$

of dimensions d_j, where $\boldsymbol{x}_0^j \in \mathbb{R}^D$, $U_{d_j}^j \in \mathbb{R}^{D \times d_j}$ has orthonormal columns, and $\boldsymbol{y} \in \mathbb{R}^{d_j}$. Assume that within each subspace S_j the data is distributed around m_j cluster centers $\{\mu_{jk} \in \mathbb{R}^D\}_{j=1\ldots n}^{k=1\ldots m_j}$.

   (a) Assume that n, d_j and m_j are known and propose a clustering algorithm similar to K-means and K-subspaces to estimate the model parameters $\boldsymbol{x}_0^j$, U_d^j, $\boldsymbol{y}_i^j$ and μ_{jk}, and the segmentation of the data according to the $\sum_{j=1}^n m_j$ 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_j and m_j are unknown. How would you modify the cost function of part a)?

4. **Implementation of Subspace Clustering Algorithms**

   (a) Investigate the function `kmeans` in MATLAB, that implements the K-means algorithm for clustering data distributed around n cluster centers.

   (b) Write a function to cluster data drawn from n subspaces using the K-Subspaces algoritm. The format of the function should be

Function `[group,x0,U,Y] = ksubspaces(x,n,d,group0)`

Parameters

`x`	$D \times N$ matrix whose columns are the data points
`n`	number of groups
`d`	$1 \times n$ vector containing the dimension of each subspace
`group0`	$1 \times N$ vector with initial group membership of each point (optional argument)

Returned values

`group`	$1 \times N$ vector with group membership of each point
`x0`	$D \times n$ matrix whose columns are the cluster centers
`U`	n-dimensional structure whose j-th entry, `U(j).matrix`, is the matrix $U_{d_j}^j \in \mathbb{R}^{D \times d_j}$, whose columns form a basis for the j-th subspace
`Y`	n-dimensional structure whose j-th entry, `Y(j).matrix`, is the low-dimensional representations $Y^j \in \mathbb{R}^{d_j \times N_j}$

Description

Computes the clustering of points using K-Subspaces.

If `group0` is not given, then the algorithm should generate a random initial segmentation.

   (c) Implement the `GPCA` algorithm for hyperplanes using the following format

Function `[b,group] = gpca(x,n)`

Parameters

`x`	$D \times N$ matrix whose columns are the data points
`n`	number of groups

Returned values

`b`	$D \times n$ matrix whose columns are the normal vectors
`group`	$1 \times N$ vector with group membership of each point

Description

Hyperplane clustering using GPCA.

   (d) Generate data with 100 points in the XY plane and 100 points in the YZ plane. For example,
   `x = [randn(2,100);zeros(1,100)]; x = [x [zeros(1,100); randn(2,100)]];`
   Add Gaussian noise to the data. For example `x = x + 0.01*randn(3,200);`
   Plot the data with different colors for the two different groups. Run the K-Subspaces algorithm with random initialization. Report percentage of misclassified points. Run the GPCA algorithm and report percentage of misclassified points. Run the K-subspaces algorithm starting from the segmentation of GPCA and report the percentage of misclassified points.