HW4 - Dimensionality Reduction and Clustering

14/12/2017

Submission date: 31/12/17

- Submission is individual or in pairs.
- Submit a ZIP file containing all your files(including txt/csv files needed to run the python code) named with 9 digits of your ID. Submission Example: '200567989.zip'
- Submit your code as a 'Jupyter/IPython notebook', after you zip your files extract them into a new folder and make sure there are no runtime errors. You are allowed to submit questions which do not involve code as you wish (i.e, Jupyter notebook/PDF/word). These files should be contained in the zip file as well.
- Use python version 3.6/2.7

1 PCA

1.1 Statistical PCA

In the following question we suggest consulting the lecture notes and Vidal's Book, 'Generalized Principal Component Analysis'.
Let $x \in \mathbb{R}^2$ be a random vector. Let $\mu_x = E[x] \in \mathbb{R}^D$ and $\Sigma_x = E[(x-\mu_x)(x-\mu_x)^T] \in \mathbb{R}^{D\times D}$(why?) be, respectively, the mean and the covariance of X. Define the principal components of x as the random variables $y_i = u_i^T x + a_i \in \mathbb{R}, i \in \{1,..,d\}$, where $d \leq D$, and $u_i \in \mathbb{R}^D$ is a unit-norm vector. We choose y_i to be zero-mean uncorrelated random variables by choosing a_i and u_i in a proper way in the following. Furthermore, assume that $\mathrm{Var(y_1)} \geq \mathrm{Var}(y_2) \geq .. \geq \mathrm{Var}(y_d)$ and that $Tr(\Sigma_y)$ attains its maximal possible value. Lastly, assume that the eigenvalues of Σ_x are distinct.

Find the values of a_i and u_i. Show and answer the following:

1. $\Sigma_y = V^T \Sigma_x V$, and find V in terms of the vectors $\{u_i\}$.
2. $a_i = -u_i^t \mu_x, \forall i \in \{1, .., d\}$.
3. What is the meaning of $Tr(\Sigma_y)$?
4. The trace of $\Sigma_y$is maximized by a matrix, V, whose columns are the first d unit eigenvectors of Σ_x. Find the matrix V. Hint: Is the matrix Σ_x is similar to a diagonal matrix (or can you diagonlize Σ_x)? .
5. $u_i^T u_j = 0$ for all $i \neq j$, and u_i is the eigenvector of Σ_x corresponding to its its largest eigenvalue.

1.2 Equivalence of Statistical and Geometric View of PCA

Explain in your own words the difference of the statistical and geometric view point. Specifically,

- Discuss the difference in optimization criterions according to these different view points.
- Explain the redundancy in the MSE solution and explain why it is absent in the variance max formulation.

2 KPCA

1. Show explicitly that the following functions are positive semidefinite kernels (i.e, find the feature space these kernels represent):
   - $k(x, y) = \phi(x)^T \phi(y)$ for some embedding function $\phi : \mathbb{R}^D \to \mathbb{R}^M$.
   - $k(x, y) = (x^T y)^n$ for $n \in \mathbb{N}$. What is the dimension of the feature space?
   - $k(x, y) = \exp(-\frac{||x-y||^2}{2\sigma^2})$ for $\sigma^2 > 0$. What is the dimension of the feature space?. Hint: Taylor expand.
2. Let $\{x_i \in \mathbb{R}^D\}_{i=1}^N$ be a set of points that you believe live in a manifold of dimension d. Imagine you have applied PCA and KPCA with kernel κ. Assume now that you are given a new point $x \in \mathbb{R}^D$. You wish to find its corresponding point $y \in \mathbb{R}^d$ according to each of the two methods. How would you compute y *without* applying PCA or KPCA from scratch

to the $N+1$ points? Under what conditions is the solution you propose equivalent to applying PCA and KPCA $N+1$ points?

3. Show that the conventional linear PCA algorithm is recovered as a special case of kernel PCA. i.e, if we choose the linear kernel function given by $k(x,y) = x^T y$ we get the PCA algorithm.

3 Code

1. Implement PCA, and KPCA algorithms (do not use sklearn library, write the code yourself!). Consult the PCA implementation we saw in the tutorial.

   The PCA function should have the form `PCA(X,d)`.

   The KPCA function should have the form `KPCA(X,d,kernel,*args)`.

   Here $X \in \mathbb{R}^{D\times N}$ is the data matrix with N data samples and D features, d is the final dimension of each data point, *kernel* is a function which receives x, y and more parameters, all captured in the $*args$ input (e.g, the Gaussian kernel function will have the form `GaussianKernel(x,y,`σ^2`)`, and $*args$, in this case will be the value of σ^2).

   The $*args$ input is a common syntax in python and it is worthwhile understanding it before starting to write the code.

2. Create a dataset of 1000 points using make_s_curve. Use the default definitions. Apply the PCA and KPCA algorithm to dimension $d = 2$. Use the Gaussian kernel and a second (nonlinear) kernel of your choice. Justify your choice for the choice of the Gaussian width, σ^2.

   Plot the original dataset (in 3D), and its representation in 2D obtain by the above (i.e, show the results for PCA and two different kernels for the KPCA algorithm). Discuss the results.