COMS30035, Machine learning: Kernel PCA

James Cussens
james.cussens@bristol.ac.uk

Department of Computer Science, SCEEM
University of Bristol

October 29, 2020

Agenda

- Kernel PCA

In the beginning there is the sample covariance matrix

- Recall that principal components are eigenvectors of the sample covariance matrix.
- If we standardise the data to have zero mean then this matrix has the following form:

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^T \tag{1}$$

and in feature space we have the $M \times M$ sample covariance matrix:

$$\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T \tag{2}$$

- Note that $k(\mathbf{x}_n, \mathbf{x}_n) = \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_n) \neq \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^T$
- The trick with kernel PCA is to avoid dealing with (eigenvectors/eigenvalues of) $\mathbf{C}$ directly.

Solving kernel PCA

- Let M be the dimension of the feature space.
- Instead of finding eigenvectors (of some sample covariance matrix) we find N-dimensional vectors $\mathbf{a}_i$ (so *nonparametric*) for $i = 1, \ldots, M$.
- $\mathbf{a}_i$ [and $\phi(\mathbf{x}_n)$ the data values in feature space] determines $\mathbf{v}_i$ the ith eigenvector in feature space.
- We can solve for $\mathbf{a}_i$:

$$\mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i \tag{3}$$

$$\lambda_i N \mathbf{a}_i^T \mathbf{a}_i = 1 \tag{4}$$

Projecting with kernel PCA

Once we have the $\mathbf{a}_i$ we can project onto the principal components in feature space just using kernel values, since:

$$\phi(\mathbf{x})^T \mathbf{v}_i = \sum_{n=1}^{N} a_{in} k(\mathbf{x}, \mathbf{x}_n) \tag{5}$$

Properties of kernel PCA

- Since M is typically much larger than D (=dimension of original data space),
- we can get more principal components than D, but
- there can be at most N non-zero eigenvalues (since we are doing an eigenvector expansion of the $N \times N$ Gram matrix $\mathbf{K}$).

Kernel PCA problems

- Previous equations assumed that the data had zero mean in feature space.
- Since this will rarely be true we have to use a 'shifted' Gram matrix $\tilde{\mathbf{K}}$ (see [Bis06, (12.85)] for details).
- $\tilde{\mathbf{K}}$ is $N \times N$ which can be too big, so approximation have to be used.
- We can't get an approximation by projections (like normal PCA).
- ϕ maps the D-dimensional data onto a D-dimensional manifold in M-dimensional feature space, but the projections in feature space won't lie in this manifold.

Now do the quiz!

Yes, please do the quiz for this lecture on Blackboard!


Christopher M. Bishop.
Pattern Recognition and Machine Learning.
Springer, 2006.