

Linking Spectral Clustering and K-PCA

Henry Bourne

24/02/2023

bristol.ac.uk

Contents

K-means Clustering

- What is it?

- Where does it fall short?

Spectral Clustering

Kernel PCA

Learning Eigenfunctions of similarity

- The General Problem

- Learning the Leading Eigenfunctions

- Spectral Clustering \subset General Problem

- Kernel PCA \subset General Problem

- Spectral Clustering = Kernel PCA

Remarks

bristol.ac.uk

K-means Clustering

- ✿ Given a set of data points $X = \{x_1, \dots, x_n\}$, where each $x_i \in \mathbb{R}^d$, find k clusters of data points.
- ✿ The clusters are defined by the k cluster centers μ_1, \dots, μ_k , where each $\mu_j \in \mathbb{R}^d$.
- ✿ Where we assign each data point x_i to the cluster with the closest cluster centre:

$$S_j = \{x_i \in X : \|x_i - \mu_j\| < \|x_i - \mu_l\| \text{ for all } l \neq j\}, j \in \{1, \dots, k\}$$

- ✿ The goal is to find the cluster centers μ_1, \dots, μ_k that minimise the within-cluster sum of squares:

$$\sum_{j=1}^k \sum_{i=1}^n \mathbb{I}_{\{x_i \in S_j\}} \|x_i - \mu_j\|^2$$

Where k-means Clustering Falls Short

- ✿ The clusters are defined by the cluster centers μ_1, \dots, μ_k , which are fixed points.
- ✿ This means that the clusters are not allowed to change shape.
- ✿ This is a problem when the ("true") clusters are not spherical.
- ✿ If the data is not linearly separable then it will be impossible to achieve a good clustering.

Contents

K-means Clustering

- What is it?

- Where does it fall short?

Spectral Clustering

Kernel PCA

Learning Eigenfunctions of similarity

- The General Problem

- Learning the Leading Eigenfunctions

- Spectral Clustering \subset General Problem

- Kernel PCA \subset General Problem

- Spectral Clustering = Kernel PCA

Remarks

bristol.ac.uk

Spectral Clustering

- ✳ Spectral clustering works by transforming the data into a new space where the clusters are more clearly defined and then performing k-means clustering in this new space.
- ✳ Spectral clustering originally comes from the field of graph theory, where the aim is to identify communities of nodes in a graph from the edges connecting them.
- ✳ This can be approximately solved by finding the eigenvectors of the Laplacian matrix of the graph.
- ✳ We will skip over the details of where Spectral clustering comes from (which can be found in the lecture notes) and instead focus on how we perform Spectral clustering on data.

Spectral Clustering: The Gram Matrix

We obtain the transformation of the data as follows, first we find the symmetric semi positive definite Gram matrix, M , defined as:

$$M_{i,j} = K(x_i, x_j)$$

where K is a kernel function (we will talk more about this shortly). Introducing some notation for the row sums:

$$D_i = \sum_j M_{i,j}$$

we then normalize the Gram matrix to obtain the following:

$$\hat{M}_{i,j} = \frac{M_{i,j}}{\sqrt{D_i D_j}} \tag{1}$$

Spectral Clustering: The Kernel

Now let's talk about kernels. Assuming we know the definition of a kernel, let's introduce the kernel that we will use in this lecture, **the Gaussian kernel**:

$$K(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma^2}\right) \quad (2)$$

where σ is a hyperparameter that controls the width of the Gaussian, it is called the **bandwidth** and its choice can have a significant impact on the performance of the algorithm.

Spectral Clustering: Normalizing

In eq. (1) we defined the normalized Gram matrix, the corresponding "normalized kernel" for this Gram matrix is:

$$\hat{K}(x, y) = \frac{K(x, y)}{\sqrt{E_x[K(x, x)] E_y[K(y, y)]}} \quad (3)$$

where expectations are over the empirical distribution of the data. Note, this is in fact a positive definite kernel as shown in [1].

This normalized kernel will be popping up again later.

Spectral Clustering: Embedding

Now all is left is to compute the principal m eigenvectors of the normalized Gram matrix, \hat{M} , and use these as the new features for the data:

$$\hat{M} \alpha_k = \lambda_k \alpha_k$$

where α_k are the eigenvectors and λ_k are the eigenvalues.

Let A be the matrix where each row corresponds to one of the first m eigenvectors, then $A \in \mathbb{R}^{m \times n}$. Then for each data point x_i its embedding is the i th column of A .

For further information on where exactly this algorithm comes from and how it relates to the graph min-cut problem please refer to [2].

bristol.ac.uk

Contents

K-means Clustering

- What is it?

- Where does it fall short?

Spectral Clustering

Kernel PCA

Learning Eigenfunctions of similarity

- The General Problem

- Learning the Leading Eigenfunctions

- Spectral Clustering \subset General Problem

- Kernel PCA \subset General Problem

- Spectral Clustering = Kernel PCA

Remarks

bristol.ac.uk

Kernel PCA

Kernel PCA (or K-PCA) generalizes PCA to be able to perform non-linear transformations of the data. When performing K-PCA we project the data into a new feature space using some transform ϕ , we then use a kernel trick.

Let's say we have some transform function ϕ , we would like to perform PCA on the data in this new feature space, $\phi(x)$. To do this we need to find the covariance matrix of the data:

$$C = E_x [\phi(x)\phi(x)^T] \quad (4)$$

Once we've found the covariance matrix we need to find the eigenvectors and eigenvalues of this matrix:

$$Cv_k = \lambda_k v_k$$

Kernel PCA: The Kernel

Using the same kernel K as before (eq. (2)) we will as before first "normalise" the kernel:

$$\hat{K}(x, y) = K(x, y) - E_x[K(x, y)] - E_y[K(x, y)] + E_x[E_y[K(x, y)]] \quad (5)$$

this means the feature space points now have an expected value of zero (under the empirical distribution of the data). A derivation of this "normalized" kernel is provided in [1].

The Gram matrix associated with this kernel is therefore normalized, again we will denote this normalized gram matrix as \hat{M} .

Kernel PCA: Gram and Covariance Matrices

As we mentioned earlier we need to find the eigenvectors and eigenvalues of the covariance matrix defined in eq. (4). We can actually avoid doing this by finding the eigenvectors and eigenvalues of the Gram matrix. From Corollary 4.1 in the lecture notes we have:

$$\lambda_k = \frac{\gamma_k}{n},$$
$$v_k = \frac{\phi(X)^T \alpha_k}{\gamma_k^{1/2}}$$

(Note that our definition of \hat{K} is the same as that of K in the lecture notes, hence it's suitable to use this corollary) From this corollary we can see that we can completely avoid computing the covariance matrix and its eigenvalues and eigenvectors.

Kernel PCA: Training

So all we need to do to train a K-PCA model is solve the following for its eigenvectors and eigenvalues:

$$\hat{M} \alpha_k = \gamma_k \alpha_k$$

Kernel PCA: Test Points Projection

Then to project a point x on the k -th eigenvector of the covariance matrix (Not Gram matrix!) we compute:

$$\pi_k(x) = v_k \cdot \hat{\phi}(x) = \frac{1}{\gamma_k} \sum_i \alpha_{ki} \hat{K}(x_i, x)$$

where $\hat{\phi}(x)$ is the centered version of $\phi(x)$, and α_k is the normalization factor.

Contents

K-means Clustering

- What is it?

- Where does it fall short?

Spectral Clustering

Kernel PCA

Learning Eigenfunctions of similarity

- The General Problem

- Learning the Leading Eigenfunctions

- Spectral Clustering \subset General Problem

- Kernel PCA \subset General Problem

- Spectral Clustering = Kernel PCA

Remarks

bristol.ac.uk

The General Problem

Consider the following Hilbert space, \mathcal{H} :

A set of real valued functions in \mathbb{R}^d equipped with an inner product defined with a density $p(x)$:

$$\langle f, g \rangle := \int f(x)g(x)p(x)dx$$

Note, this also defines a norm over functions:

$$\|f\|^2 := \langle f, f \rangle$$

The General Problem

- ✿ Let K be the linear operator corresponding to the kernel $K(x, y)$.
- ✿ The eigenfunctions of the linear operator K are defined by the solutions of:

$$Kf_k = \lambda_k f_k$$

- ✿ where $f \in \mathcal{H}$, $\lambda_k \in \mathbb{R}$ and:

$$(Kf_k)(x) := \int K(x, y)f_k(y)p(y)dy$$

The General Problem

- ✿ By Mercer's theorem, K can be expanded in terms of an orthonormal basis formed by its eigenfunctions:

$$K(x, y) = \sum_{k=1}^{\infty} \lambda_k f_k(x) f_k(y) \quad (6)$$

- ✿ (with $|\lambda_1| \geq |\lambda_2| \geq \dots$ by convention)
- ✿ Because we choose the eigenfunctions to be orthonormal, we have:

$$\langle f_k, f_l \rangle = \delta_{k,l}$$

Learning the Leading Eigenfunctions

- ✂ How do we actually find these eigenfunctions?
- ✂ A method for finding them is given in [1] in proposition 1 and 2.
- ✂ We won't cover it here for the sake of brevity.

Spectral Clustering \subset General Problem

The following proposition tells us that spectral clustering is the same as finding the principal eigenfunctions if we chose $p(x)$ to be the empirical distribution of the data:

Proposition 1 (proposition 3 from [1])

If we choose $p(x)$ to be the empirical distribution of the data, then:

$$A_{ik} = f_k(x_i) \tag{7}$$

where A_{ik} is the embedding obtained with spectral clustering and f_k is the k th principal eigenfunction of the kernel $K(x, y)$.

Note: here K could be a normalized kernel such as that we defined for spectral clustering earlier (eq. (3))

Proof: proposition 1

First let's introduce a lemma that will help us prove the proposition:

Lemma 1 (Proposition 1 in [1])

The principal eigenfunction of the linear operator corresponding to kernel K is the norm-1 function f that minimizes the following reconstruction error:

$$E(f, \lambda) = \int (K(x, y) - \lambda f(x)f(y))^2 p(x)p(y) dx dy$$

Proof: proposition 1

This has a solution that satisfies:

$$\int K(x, y)f(y)p(y)dy = \lambda f(x) \quad (8)$$

where λ is the largest eigenvalue and therefore $f(=f_1)$ is the principal eigenfunction.

Proof: proposition 1

Substituting into eq. (8) the empirical density and considering the values of x for each of our data points, we have:

$$\frac{1}{n} \sum_j K(x_i, x_j) f_1(x_j) = \lambda_1 f_1(x_i)$$

Letting $u_j = f(x_j)$ and $M_{ij} = K(x_i, x_j)$, then the above can be written as:

$$Mu = n\lambda u$$

This is the same as the eigenvalue problem we have to solve for spectral clustering up to scaling the eigenvalue by n .

Proof: proposition 1

Therefore for the principal eigenvector we have:

$$A_{i1} = f_1(x_i)$$

How about the other eigenvalues? let's first introduce what we will call the k -th **residual kernel** to simplify notation:

$$K_k(x, y) = K(x, y) - \sum_{i=1}^k \lambda_i f_i(x) f_i(y) \quad (9)$$

Proof: proposition 1

To obtain the principal m eigenvectors we simply consider the k -th residual kernel for $k \in \{1, \dots, m\}$ and solve the eigenvalue problem for that kernel.

(ie. subbing into eq. (8) the empirical density and f_k, λ_k)

We then have: $A_{ik} = f_k(x_i)$ for $k \in \{1, \dots, m\}$.

Kernel PCA \subset General Problem

The following proposition tells us that kernel PCA is the same as finding the principal eigenfunctions if we chose $p(x)$ to be the empirical distribution of the data:

Proposition 2 (proposition 4 from [1])

Let $\pi_k(x)$ be the projection of x onto the k th principal component obtained by KPCA where the Hilbert space inner product weighting function $p(x)$ is the empirical density. Then:

$$\pi_k(x) = \lambda_k f_k(x)$$

where λ_k and f_k are k -th leading eigenvalue and eigenfunction of \hat{K} (as defined in eq. (5)).

Proof: proposition 2

Let's start with the eigenfunction equation:

$$\hat{K} f_k = \lambda_k f_k$$

$$\iff \hat{K} \hat{K} f_k = \lambda_k \hat{K} f_k$$

$$\iff \int \hat{K}(x, y) \int \hat{K}(y, z) f_k(z) p(z) p(y) dz dy = \lambda_k \int \hat{K}(x, y) f_k(y) p(y) dy$$

$$\iff \int f_k(z) \left(\int \hat{K}(x, y) \hat{K}(y, z) p(y) dy \right) p(z) dz = \lambda_k \int \hat{K}(x, y) f_k(y) p(y) dy$$

Proof: proposition 2

Now plugging in our definition of $\hat{K}(x, y) = \sum_i \hat{\phi}_i(x) \hat{\phi}_i(y)$ (where $\hat{\phi}(x)$ is the centered version of $\phi(x)$):

$$\begin{aligned} &\iff \int f_k(z) \left(\int \sum_i \hat{\phi}_i(x) \hat{\phi}_i(y) \sum_j \hat{\phi}_j(y) \hat{\phi}_j(z) p(y) dy \right) p(z) dz \\ &= \lambda_k \int \sum_i \hat{\phi}_i(x) \hat{\phi}_i(y) f_k(y) p(y) dy \end{aligned}$$

Recall that the covariance matrix, C , of data x when projected in feature space $\hat{\phi}(x)$ is given by:

$$C_{ij} = E(\hat{\phi}(x) \hat{\phi}(x)^T)$$

therefore:

bristol.ac.uk

$$C_{ij} = \int \hat{\phi}_i(y) \hat{\phi}_j(y) p(y) dy$$

Proof: proposition 2

Now plugging this in and pulling sums out of integrals we have:

$$\begin{aligned} \sum_i \hat{\phi}_i(x) \sum_j C_{ij} \int \hat{\phi}_j(z) f_k(z) p(z) dz &= \lambda_k \sum_i \hat{\phi}_i(x) \int f_k(y) \hat{\phi}_i(y) p(y) dy \\ \iff \hat{\phi}(x) \cdot (C \langle f_k, \hat{\phi} \rangle) &= \hat{\phi}(x) \cdot (\lambda_k \langle f_k, \hat{\phi} \rangle) \end{aligned}$$

therefore we must have:

$$C \langle f_k, \hat{\phi} \rangle = \lambda_k \langle f_k, \hat{\phi} \rangle$$

letting $v_k = \langle f_k, \hat{\phi} \rangle$ we have:

$$C v_k = \lambda_k v_k$$

so v_k is the k-th eigenvector of the covariance matrix, C .

bristol.ac.uk

Proof: proposition 2

Finally the projection of x onto the k -th principal component is given by:

$$\begin{aligned}\pi_k(x) &= v_k \cdot \hat{\Phi}(x) \\ &= \left(\int f_k(y) \hat{\Phi}(y) p(y) dy \right) \cdot \hat{\Phi}(x) \\ &= \int f_k(y) \hat{\Phi}(y) \cdot \hat{\Phi}(x) p(y) dy \\ &= \int f_k(y) \hat{K}(x, y) p(y) dy \\ &= \lambda_k f_k(x)\end{aligned}$$

Hence the proposition is proven.

bristol.ac.uk

Spectral Clustering = Kernel PCA

Combining the results of proposition 1 and proposition 2 we have that spectral clustering is the same as kernel PCA up to scaling of the eigenvalues and a different normalization of the kernel (eq. (3) vs eq. (5)).

Contents

K-means Clustering

- What is it?

- Where does it fall short?

Spectral Clustering

Kernel PCA

Learning Eigenfunctions of similarity

- The General Problem

- Learning the Leading Eigenfunctions

- Spectral Clustering \subset General Problem

- Kernel PCA \subset General Problem

- Spectral Clustering = Kernel PCA

Remarks

bristol.ac.uk

Remarks

What do we gain from knowing this link between spectral clustering and K-PCA?

- ✂ We can generalize an embedding to a mapping, eg. for spectral clustering we can embed new points.
- ✂ Using the eigenfunction method we can change the probability distribution of the data, $p(x)$, eg. we could use a smoothed version.
- ✂ We don't necessarily have to compute and store the Gram matrix if we use the eigenfunction method.
- ✂ They also introduce an unsupervised stochastic method for learning eigenfunctions in [1], using this method we can actually recursively build higher and higher level representations of the data, similarly to what happens in deep learning.

Contents

K-means Clustering

- What is it?

- Where does it fall short?

Spectral Clustering

Kernel PCA

Learning Eigenfunctions of similarity

- The General Problem

- Learning the Leading Eigenfunctions

- Spectral Clustering \subset General Problem



- Kernel PCA \subset General Problem

- Spectral Clustering = Kernel PCA

Remarks

bristol.ac.uk

Bibliography

-  Yoshua Bengio et al. *Learning eigenfunctions of similarity: linking spectral clustering and kernel PCA*. Tech. rep. Technical Report 1232, Departement d'Informatique et Recherche Operationnelle . . ., 2003.
-  Andrew Ng, Michael Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems* 14 (2001).