

## Kernel PCA

---

Advanced Institute for Artificial Intelligence – AI2

<https://advancedinstitute.ai>

*Kernel PCA* (KPCA) é uma técnica que busca generalizar PCA visando realizar **redução não linear de dimensionalidade**. Lembrando que PCA assume que os dados estão em um espaço Euclidiano, ou seja, linear. Na hipótese que os dados estão em um espaço não linear, PCA não consegue garantir a sua otimalidade.

Seja, então, um conjunto de dados  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$  tal que  $\mathbf{x}_i \in \mathbb{R}^n$  representa uma amostra no espaço de características original. A ideia do KPCA é fazer uso de uma função de mapeamento não linear, também conhecida por *kernel*,  $\phi(\mathbf{x}) \in \mathbb{R}^{n'}$ , tal que  $n' > n$ , visando mapear os dados para um espaço de maior dimensão. Geralmente, esses mapeamentos são custosos pois envolvem a projeção das amostras nestes espaços com maior dimensão. No entanto, podemos contornar isso com o *kernel trick*, em que podemos **calcular o produto interno** no espaço de alta dimensão **sem a necessidade** de projeção dos dados.

Desta forma, KPCA realiza um aumento temporário da dimensionalidade do espaço para depois realizar a sua redução. **A ideia é incorporar não linearidades durante o processo de redução de dimensionalidade.**

A primeira hipótese do KPCA é assumir que a média amostral após o mapeamento para o espaço de maior dimensão vale zero, ou seja:

$$\boldsymbol{\mu}' = \frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_i) = 0, \quad (1)$$

em que  $\boldsymbol{\mu}' \in \mathbb{R}^{n'}$ . Neste caso, KPCA assume que a média amostral está situada na origem do espaço de dados.

Já a matriz de covariância  $\Sigma' \in \mathbb{R}^{n' \times n'}$  é dada por:

$$\Sigma' = \frac{1}{m} \sum_{i=1}^m (\phi(\mathbf{x}_i) - \cancel{\mu'}^0)(\phi(\mathbf{x}_i) - \cancel{\mu'}^0)^T = \frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T. \quad (2)$$

Temos que os autovetores  $\mathbf{v}_k$  da matriz  $\Sigma$  são dados por:

$$\Sigma' \mathbf{v}_k = \lambda \mathbf{v}_k, \quad \forall k = 1, 2, \dots, n'. \quad (3)$$

Existe um teorema bastante importante que nos diz que os autovetores de  $\Sigma$  podem ser escritos como uma combinação linear das características, ou seja:

$$\mathbf{v}_k = \sum_{i=1}^m \alpha_{ki} \phi(\mathbf{x}_i). \quad (4)$$

Assim sendo, para calcularmos os autovetores da matriz de covariância  $\Sigma'$ , basta sabermos os vetores  $\alpha_k$ . Substituindo-se a Equação 2 na Equação 3, temos que:

$$\Sigma' \mathbf{v}_k = \frac{1}{m} \underbrace{\sum_{i=1}^m \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T}_{\Sigma'} \mathbf{v}_k = \lambda \mathbf{v}_k. \quad (5)$$

A Equação 5 implica em:

$$\mathbf{v}_k = \frac{1}{m\lambda} \sum_{i=1}^m \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \mathbf{v}_k = \frac{1}{m\lambda} \sum_{i=1}^m [\phi(\mathbf{x}_i)^T \mathbf{v}_k] \phi(\mathbf{x}_i) = \sum_{i=1}^m \alpha_{ki} \phi(\mathbf{x}_i), \quad (6)$$

em que  $\alpha_{ki} = \frac{1}{m\lambda} \phi(\mathbf{x}_i)^T \mathbf{v}_k$ . Agora, substituindo-se a Equação 6 na Equação 5, temos que:

$$\frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T \left( \sum_{j=1}^m \alpha_{kj} \phi(\mathbf{x}_j) \right) = \lambda \sum_{j=1}^m \alpha_{kj} \phi(\mathbf{x}_j). \quad (7)$$

Reescrevendo a Equação 7, temos que:

$$\frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_i) \left( \sum_{j=1}^m \alpha_{kj} \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \right) = \lambda \sum_{j=1}^m \alpha_{kj} \phi(\mathbf{x}_j). \quad (8)$$

Fazendo uso do *kernel trick*, ou seja,  $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$  temos que:

$$\frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_i) \left( \sum_{j=1}^m \alpha_{kj} K(\mathbf{x}_i, \mathbf{x}_j) \right) = \lambda \sum_{j=1}^m \alpha_{kj} \phi(\mathbf{x}_j). \quad (9)$$

Multiplicando ambos lados da Equação 9 por  $\phi(\mathbf{x}_l)^T$ , temos que:

$$\frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_l)^T \phi(\mathbf{x}_i) \left( \sum_{j=1}^m \alpha_{kj} K(\mathbf{x}_i, \mathbf{x}_j) \right) = \lambda \sum_{j=1}^m \alpha_{kj} \phi(\mathbf{x}_l)^T \phi(\mathbf{x}_j). \quad (10)$$

Usando novamente o *kernel trick*, temos que:

$$\frac{1}{m} \sum_{i=1}^m K(\mathbf{x}_l, \mathbf{x}_i) \left( \sum_{j=1}^m \alpha_{kj} K(\mathbf{x}_i, \mathbf{x}_j) \right) = \lambda \sum_{j=1}^m \alpha_{kj} K(\mathbf{x}_l, \mathbf{x}_j). \quad (11)$$

Temos que nossa formulação está, agora, escrita em função da matriz de *kernel*.



Podemos expressar a Equação 11 em sua notação matriz-vetor, como segue:

$$\mathbf{K}^2 \boldsymbol{\alpha}_k = (\lambda m) \mathbf{K} \boldsymbol{\alpha}_k. \quad (12)$$

Multiplicando-se ambos lados por  $\mathbf{K}^{-1}$ , temos que:

$$\mathbf{K} \boldsymbol{\alpha}_k = (\lambda m) \boldsymbol{\alpha}_k. \quad (13)$$

Quem são os  $\boldsymbol{\alpha}_k$ ? São os **autovetores** da matriz de *kernel*!

Para uma dada amostra  $\mathbf{x}$ , calculamos a sua projeção na  $k$ -ésima componente da seguinte forma:

$$\hat{\mathbf{x}} = \phi(\mathbf{x})^T \mathbf{v}_k = \underbrace{\sum_{i=1}^m \alpha_{ki} \phi(\mathbf{x}_i)}_{\mathbf{v}_k} \phi(\mathbf{x})^T = \sum_{i=1}^m \alpha_{ki} K(\mathbf{x}, \mathbf{x}_i). \quad (14)$$

Basicamente, a projeção de uma amostra  $\mathbf{x}$  na  $k$ -ésima componente resume-se a realizar o seu mapeamento para um espaço de maior dimensão com  $\phi(\mathbf{x})$  e depois projetar em  $\mathbf{v}_k$ . **A vantagem do *kernel trick* é que não precisamos calcular explicitamente  $\phi(\mathbf{x}_i)$ ,  $\forall i$ , basta apenas computar a matriz de *kernel* diretamente do conjunto de dados.**

Quais tipos de *kernel* podemos utilizar? Os mais empregados são:

- Polinomial:  $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d$ , em que  $c \geq 0$  é uma constante e  $d$  o grau do polinômio.
- Gaussiano (RBF):  $K(\mathbf{x}_i, \mathbf{x}_j) = \exp \left\{ -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right\}$ .

Note, também, que se o dado projetado não tiver média nula, precisamos centralizá-lo da seguinte maneira:

$$\tilde{\phi}(\mathbf{x}_i) = \phi(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^m \phi(\mathbf{x}_j). \quad (15)$$

Já a matriz *kernel* correspondente (centralizada) é dada por:

$$\begin{aligned}\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) &= \tilde{\phi}(\mathbf{x}_i)^T \tilde{\phi}(\mathbf{x}_j) = \left( \phi(\mathbf{x}_i) - \frac{1}{m} \sum_{k=1}^m \phi(\mathbf{x}_k) \right)^T \left( \phi(\mathbf{x}_j) - \frac{1}{m} \sum_{k=1}^m \phi(\mathbf{x}_k) \right) \\&= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) - \frac{1}{m} \sum_{k=1}^m \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) - \frac{1}{m} \sum_{k=1}^m \phi(\mathbf{x}_k)^T \phi(\mathbf{x}_j) \\&\quad + \frac{1}{m^2} \sum_{k=1}^m \sum_{l=1}^m \phi(\mathbf{x}_k)^T \phi(\mathbf{x}_l) \\&= K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{m} \sum_{k=1}^m K(\mathbf{x}_i, \mathbf{x}_k) - \frac{1}{m} \sum_{k=1}^m K(\mathbf{x}_k, \mathbf{x}_j) \\&\quad + \frac{1}{m^2} \sum_{k=1}^m \sum_{l=1}^m K(\mathbf{x}_k, \mathbf{x}_l).\end{aligned}\tag{16}$$

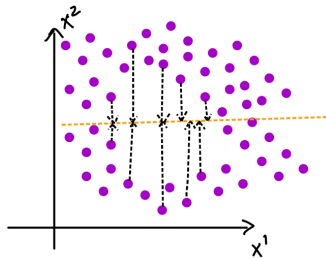
Em sua forma matricial, podemos escrever a Equação 16 como segue (matriz Gram):

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_m \mathbf{K} - \mathbf{K} \mathbf{1}_m + \mathbf{1}_m \mathbf{K} \mathbf{1}_m, \quad (17)$$

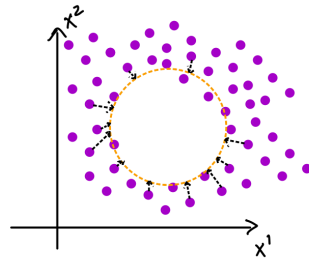
em que  $\mathbf{1}_m \in \mathbb{R}^{m \times m}$  é uma matriz com todos elementos iguais à  $1/m$ . Vejamos, agora, o algoritmo do KPCA:

- 1 Construir matriz *kernel*  $\mathbf{K}$  a partir dos dados, em que  $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$ .
- 2 Construir matriz Gram  $\tilde{\mathbf{K}}$  utilizando a Equação 17.
- 3 Utilize Equação 13 para encontrar os autovetores  $\alpha_k$  (use  $\tilde{\mathbf{K}}$ ).
- 4 Calcule os  $d$  componentes principais utilizando a Equação 14.

Exemplos de projeções utilizando PCA e KPCA.



PCA



KPCA