

Álgebra Linear para Ciência de Dados

Guilherme Tomiasi

30/10/2025

Álgebra Linear para Ciência de Dados

Métodos Kernel, Kernel PCA

Guilherme Tomiasi

30/10/2025

Feature Map

Uma maneira de transformar dados não-lineares para cumprir propriedades necessárias para a aplicação de métodos lineares.

Feature Map

Uma maneira de transformar dados não-lineares para cumprir propriedades necessárias para a aplicação de métodos lineares.

Dado um conjunto de dados $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$. Se aplicarmos uma transformação:

Feature Map

Uma maneira de transformar dados não-lineares para cumprir propriedades necessárias para a aplicação de métodos lineares.

Dado um conjunto de dados $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$. Se aplicarmos uma transformação:

$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^d \quad d \gg n$$

Feature Map

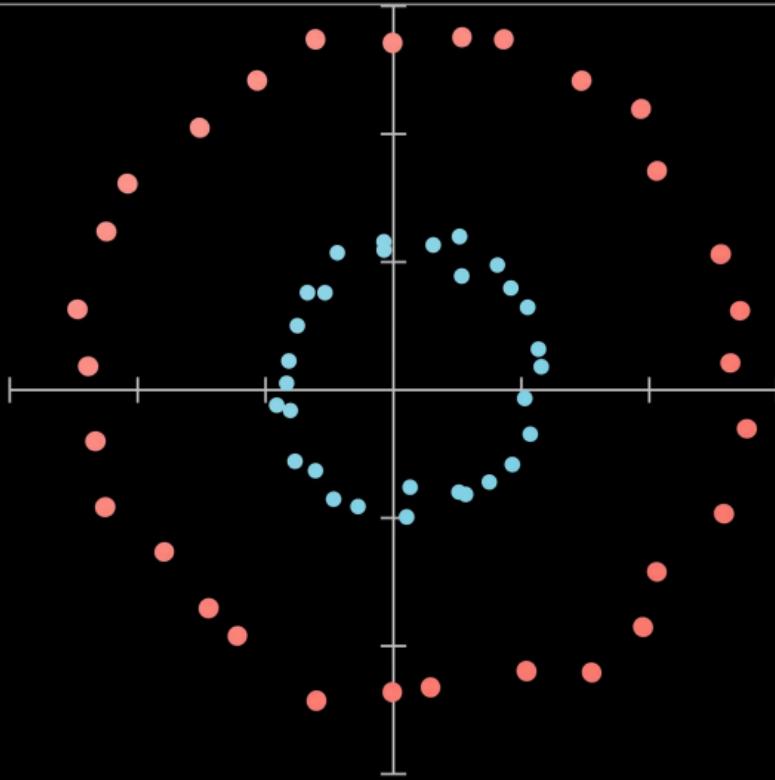
Uma maneira de transformar dados não-lineares para cumprir propriedades necessárias para a aplicação de métodos lineares.

Dado um conjunto de dados $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^n$. Se aplicarmos uma transformação:

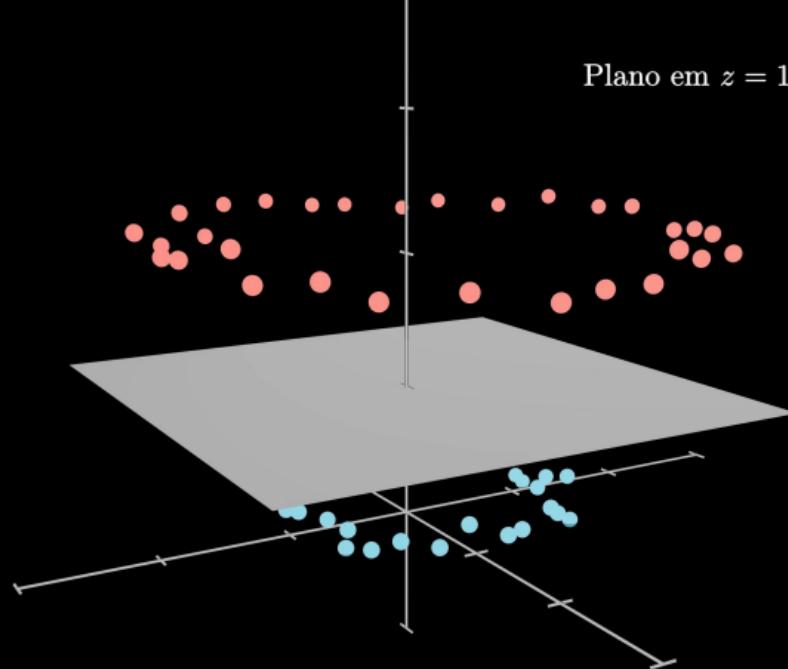
$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^d \quad d \gg n$$

Obtemos os vetores de observações $\mathbf{z}_i = \phi(\mathbf{x}_i)$. Um exemplo de aplicação a transformação de um conjunto para aplicar métodos de regressão polinomial (quadrática, cúbica) sobre a fundação de regressão linear. Outro caso comum é a transformação de um conjunto não linearmente separável em linearmente separável.

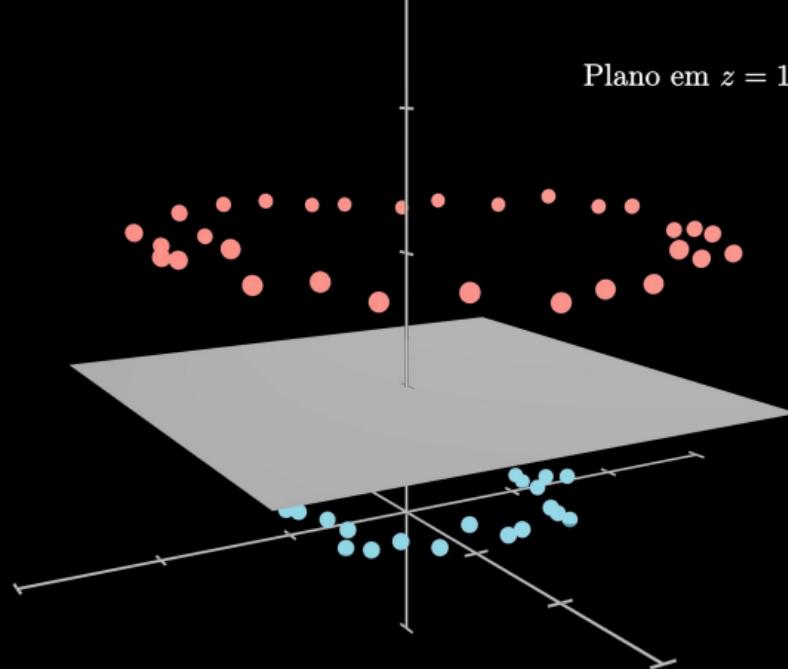
Exemplo



Plano em $z = 1$ acaba por separar as duas classes.



Plano em $z = 1$ acaba por separar as duas classes.



Problemas do *Feature Map*

O uso de *Feature Maps* é inviabilizado por alguns motivos:

Problemas do *Feature Map*

O uso de *Feature Maps* é inviabilizado por alguns motivos:

- Dificuldade de formular uma expressão de forma manual

Problemas do *Feature Map*

O uso de *Feature Maps* é inviabilizado por alguns motivos:

- Dificuldade de formular uma expressão de forma manual
- Aumento considerável do espaço vetorial utilizado (quando $d \gg n$), tornando o cálculo inviável.

Problemas do *Feature Map*

O uso de *Feature Maps* é inviabilizado por alguns motivos:

- Dificuldade de formular uma expressão de forma manual
- Aumento considerável do espaço vetorial utilizado (quando $d \gg n$), tornando o cálculo inviável.
- Dessa maneira, a forma de contornar esse problema é simplesmente **não utilizar** essa técnica específica, mas uma alternativa chamada **funções kernel**.

Kernel Function (Funções Kernel)

Definição: uma função kernel é uma função **simétrica** dada por $\mathcal{K} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$

Kernel Function (Funções Kernel)

Definição: uma função kernel é uma função **simétrica** dada por $\mathcal{K} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$

- Costuma ser utilizada para denotar a similaridade entre pares de pontos. Por exemplo, a função de distância: $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$, ou o produto interno usual: $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$, também chamado de kernel linear.

Kernel Function (Funções Kernel)

Definição: uma função kernel é uma função **simétrica** dada por $\mathcal{K} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$

- Costuma ser utilizada para denotar a similaridade entre pares de pontos. Por exemplo, a função de distância: $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$, ou o produto interno usual: $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$, também chamado de kernel linear.
- Traduzindo o conceito de *feature map* para funções kernel, chama-se de kernel associado \mathcal{K}_ϕ a expressão dada por

$$\mathcal{K}_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = \sum_{i=1}^d \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$$

Matriz kernel

Definição: dada uma função kernel, é possível construir uma matriz $K \in \mathbb{R}^{m \times m}$ onde o valor na posição (i, j) é dado por $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Como a função *kernel* é por si só simétrica, logo $K = K^T$.

Matriz kernel

Definição: dada uma função kernel, é possível construir uma matriz $K \in \mathbb{R}^{m \times m}$ onde o valor na posição (i, j) é dado por $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Como a função *kernel* é por si só simétrica, logo $K = K^T$.

- Ao utilizar o kernel linear, a matriz kernel é igual a matriz Gram.

Matriz kernel

Definição: dada uma função kernel, é possível construir uma matriz $K \in \mathbb{R}^{m \times m}$ onde o valor na posição (i, j) é dado por $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Como a função *kernel* é por si só simétrica, logo $K = K^T$.

- Ao utilizar o kernel linear, a matriz kernel é igual a matriz Gram.
- Caso definirmos a matriz de entrada após a aplicação de um *feature map*:

$$Z = \begin{bmatrix} \phi(\mathbf{x}_1)^T \\ \phi(\mathbf{x}_2)^T \\ \vdots \\ \phi(\mathbf{x}_m)^T \end{bmatrix} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \vdots \\ \mathbf{z}_m^T \end{bmatrix}$$

logo obtemos a equivalência $K = ZZ^T$, de forma que garantimos que K é semi-definida positiva.

Kernel de Mercer

Uma função kernel é chamada de *Kernel de Mercer* caso as matrizes kernel associadas a ela sejam semi-definida positivas para qualquer conjunto de vetores $\{\mathbf{x}_1, \dots, \mathbf{x}_m\} \in \mathbb{R}^n$ e qualquer $m \geq 1$

Kernel de Mercer

Uma função kernel é chamada de *Kernel de Mercer* caso as matrizes kernel associadas a ela sejam semi-definida positivas para qualquer conjunto de vetores $\{\mathbf{x}_1, \dots, \mathbf{x}_m\} \in \mathbb{R}^n$ e qualquer $m \geq 1$

- Em 1909, **James Mercer** publicou um artigo acerca de funções contínuas simétricas de “tipo positivo ou negativo”, esse artigo e seu autor dão nome a esse conjunto específico de funções kernel.

Kernel de Mercer

Uma função kernel é chamada de *Kernel de Mercer* caso as matrizes kernel associadas a ela sejam semi-definida positivas para qualquer conjunto de vetores $\{\mathbf{x}_1, \dots, \mathbf{x}_m\} \in \mathbb{R}^n$ e qualquer $m \geq 1$

- Em 1909, **James Mercer** publicou um artigo acerca de funções contínuas simétricas de “tipo positivo ou negativo”, esse artigo e seu autor dão nome a esse conjunto específico de funções kernel.
- Um kernel de Mercer deve satisfazer a seguinte expressão:

$$\mathbf{c}^T K \mathbf{c} = \sum_{i,j=1}^m c_i c_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

para qualquer conjunto de m vetores $\mathbf{x}_i \in \mathbb{R}^n$ e $\mathbf{c}_i \in \mathbb{R}^m$, para qualquer $m \in \mathbb{Z}^+$

Kernel de Mercer

Uma função kernel é chamada de *Kernel de Mercer* caso as matrizes kernel associadas a ela sejam semi-definida positivas para qualquer conjunto de vetores $\{\mathbf{x}_1, \dots, \mathbf{x}_m\} \in \mathbb{R}^n$ e qualquer $m \geq 1$

- Em 1909, **James Mercer** publicou um artigo acerca de funções contínuas simétricas de “tipo positivo ou negativo”, esse artigo e seu autor dão nome a esse conjunto específico de funções kernel.
- Um kernel de Mercer deve satisfazer a seguinte expressão:

$$\mathbf{c}^T K \mathbf{c} = \sum_{i,j=1}^m c_i c_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

para qualquer conjunto de m vetores $\mathbf{x}_i \in \mathbb{R}^n$ e $\mathbf{c}_i \in \mathbb{R}^m$, para qualquer $m \in \mathbb{Z}^+$

- Todo kernel de Mercer pode ser expresso a partir da aplicação de um *feature map*:

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \phi_i(\mathbf{x}) \phi_i(\mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

- $\alpha\mathcal{K}, \forall \alpha > 0$

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

- $\alpha\mathcal{K}, \forall \alpha > 0$
- $\mathcal{K}_1 + \mathcal{K}_2$

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

- $\alpha\mathcal{K}, \forall \alpha > 0$
- $\mathcal{K}_1 + \mathcal{K}_2$
- $\mathcal{K}_1(\mathbf{x}, \mathbf{y})\mathcal{K}_2(\mathbf{x}, \mathbf{y})$

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

- $\alpha\mathcal{K}, \forall \alpha > 0$
- $\mathcal{K}_1 + \mathcal{K}_2$
- $\mathcal{K}_1(\mathbf{x}, \mathbf{y})\mathcal{K}_2(\mathbf{x}, \mathbf{y})$
- $\mathcal{K}(\mathbf{x}, \mathbf{y})^d, \forall d \in \mathbb{Z}^+$

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

- $\alpha\mathcal{K}, \forall \alpha > 0$
- $\mathcal{K}_1 + \mathcal{K}_2$
- $\mathcal{K}_1(\mathbf{x}, \mathbf{y})\mathcal{K}_2(\mathbf{x}, \mathbf{y})$
- $\mathcal{K}(\mathbf{x}, \mathbf{y})^d, \forall d \in \mathbb{Z}^+$
- $\exp(\mathcal{K}(\mathbf{x}, \mathbf{y}))$

Kernel de Mercer

Sejam $\mathcal{K}, \mathcal{K}_1, \mathcal{K}_2$ kernels de Mercer. Logo, as seguintes expressões também representam kernels de Mercer:

- $\alpha\mathcal{K}, \forall \alpha > 0$
- $\mathcal{K}_1 + \mathcal{K}_2$
- $\mathcal{K}_1(\mathbf{x}, \mathbf{y})\mathcal{K}_2(\mathbf{x}, \mathbf{y})$
- $\mathcal{K}(\mathbf{x}, \mathbf{y})^d, \forall d \in \mathbb{Z}^+$
- $\exp(\mathcal{K}(\mathbf{x}, \mathbf{y}))$
- $F(\mathbf{x})\mathcal{K}(\mathbf{x}, \mathbf{y})F(\mathbf{y}), \forall F : \mathbb{R}^n \rightarrow \mathbb{R}$

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.
- Projeção de novo ponto x : $z_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^m \alpha_{ik} \mathcal{K}(\mathbf{x}_i, x)$.

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.
- Projeção de novo ponto x : $z_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^m \alpha_{ik} \mathcal{K}(\mathbf{x}_i, x)$.
- Escolha do kernel ([linear](#), [polinomial](#), [RBF](#)) define a geometria no espaço ϕ .

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.
- Projeção de novo ponto x : $z_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^m \alpha_{ik} \mathcal{K}(\mathbf{x}_i, x)$.
- Escolha do kernel (linear, polinomial, RBF) define a geometria no espaço ϕ .

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$$

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.
- Projeção de novo ponto x : $z_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^m \alpha_{ik} \mathcal{K}(\mathbf{x}_i, x)$.
- Escolha do kernel (linear, polinomial, RBF) define a geometria no espaço ϕ .

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + r)^n$$

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.
- Projeção de novo ponto x : $z_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^m \alpha_{ik} \mathcal{K}(\mathbf{x}_i, x)$.
- Escolha do kernel (linear, polinomial, RBF) define a geometria no espaço ϕ .

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{y}\|^2\right)$$

Kernel PCA

O Método da Análise das Componentes Principais (PCA) visa analisar as direções de maior variância, a partir de autovetores da covariância. Quando observa-se uma estrutura não-linear, é possível aplicar um *feature map* ϕ a partir do uso de funções kernel, ou seja, sem computar ϕ explicitamente.

- Construa a matriz kernel $K \in \mathbb{R}^{m \times m}$, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$.
- Centralize: $\tilde{K} = HKH$, $H = I - \frac{1}{m}\mathbf{1}\mathbf{1}^T$.
- Autodecomposição: $\tilde{K}\alpha_k = m\lambda_k\alpha_k$ com $\|\alpha_k\| = 1$.
- Projeção de novo ponto x : $z_k(x) = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^m \alpha_{ik} \mathcal{K}(\mathbf{x}_i, x)$.
- Escolha do kernel ([linear](#), [polinomial](#), [RBF](#)) define a geometria no espaço ϕ .