Techniques d’apprentissage

IFT 603-712

Méthodes à noyau

Par
Pierre-Marc Jodoin
/
Hugo Larochelle

Modélisation non linéaire

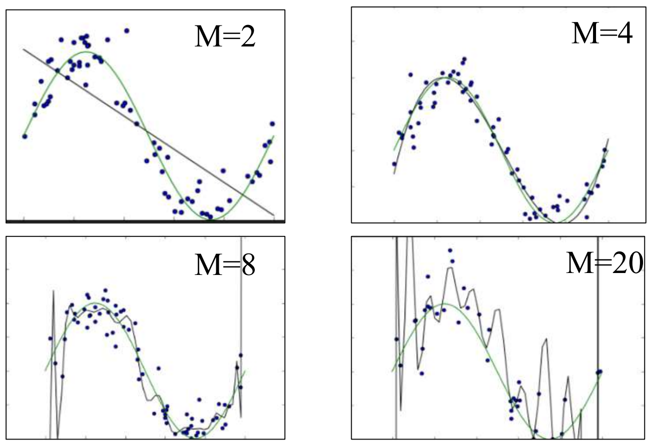
- On a vu plusieurs algorithmes basés sur des **modèles linéaires** (régression ou classification)
- Malheureusement, pas **tous les problèmes** peuvent être résolus avec un modèle linéaire
- Par contre, on peut obtenir des modèles non-linéaires à l’aide de **fonctions de base** non-linéaires

Fonctions de base polynomiales

RAPPEL

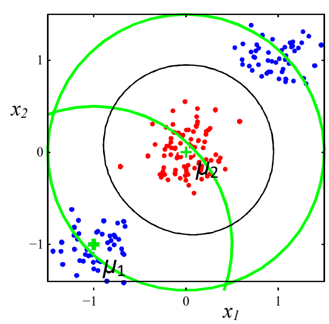
Régression polynomiale

Exemple tp1 : fonctions de bases polynomiales (*1D*) $\vec{\phi}_i(\vec{x}) = x^i$

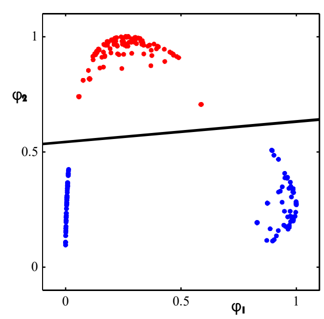


Fonctions de base gaussiennes

Segmentation avec noyau gaussien



Espace représenté par $\vec{x} = (x_1, x_2)^T$



Espace représenté par $\vec{\phi}(\vec{x}) = (\phi_1(\vec{x}), \phi_2(\vec{x}))^T$

2 fonctions de base gaussiennes

$$\phi_1(\vec{x}) = e^{-\frac{\|\vec{x}-\vec{\mu}_1\|^2}{2\sigma^2}}$$

$$\phi_2(\vec{x}) = e^{-\frac{\|\vec{x}-\vec{\mu}_2\|^2}{2\sigma^2}}$$

Méthodes à noyau

RAPPEL

Revenons au problème de la régression (**Maximum *a posteriori* – Chap 3**) :

$$\vec{w} = \arg\min_{\vec{w}} \sum_{n=1}^{N} \left(\vec{w}^T \vec{\phi}(\vec{x}_n) - t_n\right)^2 + \lambda \vec{w}^T \vec{w}$$

où

$$\vec{\phi}(\vec{x}) = (1, \phi_1(\vec{x}), \phi_2(\vec{x}), \ldots, \phi_M(\vec{x}))$$

$$\vec{w}^T = (w_0, w_1, \ldots, w_M)$$

biais **poids**

Méthodes à noyau

Revenons au problème de la régression (**Maximum *a posteriori* – Chap 3**) :

$$\vec{w} = \arg\min_{\vec{w}} \sum_{n=1}^{N} \left(\vec{w}^T \vec{\phi}(\vec{x}_n) - t_n\right)^2 + \lambda \vec{w}^T \vec{w}$$

Si on fixe le gradient par rapport à $\vec{w}$ à 0, on observe que

$$\begin{aligned}\vec{w} &= -\frac{1}{\lambda}\sum_{n=1}^{N}\left(\vec{w}^T\vec{\phi}(\vec{x}_n) - t_n\right)\vec{\phi}(\vec{x}_n) \\ &= \sum_{n=1}^{N} a_n \vec{\phi}(\vec{x}_n) \qquad \text{où } a_n = -\frac{\vec{w}^T\vec{\phi}(\vec{x}_n) - t_n}{\lambda} \in R \\ &= \Phi^T \vec{a}\end{aligned}$$

Méthodes à noyau

Revenons au problème de la régression (**Maximum *a posteriori* – Chap 3**) :

$$\vec{w} = \arg\min_{\vec{w}} \sum_{n=1}^{N} \left(\vec{w}^T \vec{\phi}(\vec{x}_n) - t_n \right)^2 + \lambda \vec{w}^T \vec{w}$$

La solution $\vec{w}$ est une **somme pondérée des entrées** $\phi(\vec{x}_n)$ dans l'ensemble d'entraînement

inconnue Connue inconnue

$$\vec{w} = \sum_{n=1}^{N} a_n \vec{\phi}(\vec{x}_n) = \Phi^T \vec{a}$$

Méthodes à noyau

inconnue Connue inconnue

$$\vec{w} = \sum_{n=1}^{N} a_n \vec{\phi}(\vec{x}_n) = \Phi^T \vec{a}$$

Idée : plutôt que d'optimiser par rapport à $\vec{w}$,

optimisons par rapport à $\vec{a}$

Méthodes à noyau

Partant de

$$J(\vec{w}) = \sum_{n=1}^{N} \left(t_n - \vec{w}^T \vec{\phi}(\vec{x}_n)\right)^2 + \lambda \vec{w}^{\mathrm{T}} \vec{w}$$

Si on remplace $\vec{w}$ par $\Phi^{\mathrm{T}} \vec{a}$ on peut démontrer qu'on obtient

$$J(\vec{a}) = \vec{a}^T \Phi \Phi^T \Phi \Phi^T \vec{a} - \vec{a}^T \Phi \Phi^T \vec{t} + \vec{t}^T \vec{t} + \lambda \vec{a}^T \Phi \Phi^T \vec{a}$$

C'est la **représentation duale** de $J(W)$

Représentation duale

On peut aussi noter que $\Phi\Phi^{\mathrm{T}} = K$ où $K_{nm} = k(\vec{x}_n, \vec{x}_m)$

Matrice de Gram

et $k(\vec{x}_n, \vec{x}_m) = \phi(\vec{x}_n)^{\mathrm{T}} \phi(\vec{x}_m)$

Noyau

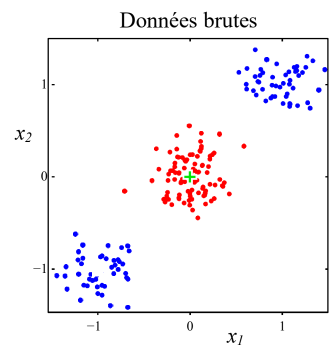
$$J(\vec{a}) = \vec{a}^T \Phi \Phi^T \Phi \Phi^T \vec{a} - \vec{a}^T \Phi \Phi^T \vec{t} + \vec{t}^T \vec{t} + \lambda \vec{a}^T \Phi \Phi^T \vec{a}$$

$$J(\vec{a}) = \vec{a}^{\mathrm{T}} K K \vec{a} - \vec{a}^{\mathrm{T}} K \vec{t} + \vec{t}^{\mathrm{T}} \vec{t} + \lambda \vec{a}^{\mathrm{T}} K \vec{a}$$

Matrice de Gram

La matrice de Gram est une **matrice *N* x *N*** contenant le **produit scalaire** entre chaque élément de l'ensemble d'entraînement.

Illustration en 2D



Espace représenté par $\vec{x} = (x_1, x_2)^{\mathrm{T}}$

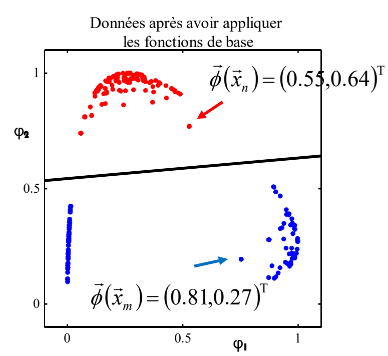
Matrice de Gram

La matrice de Gram est une **matrice *N* x *N*** contenant le **produit scalaire** entre chaque élément de l'ensemble d'entraînement.

Illustration en 2D

$$K = \begin{pmatrix} k(\vec{x}_1,\vec{x}_1) & \cdots & k(\vec{x}_1,\vec{x}_N) \\ \vdots & k(\vec{x}_n,\vec{x}_m) & \vdots \\ k(\vec{x}_N,\vec{x}_1) & \cdots & k(\vec{x}_N,\vec{x}_N) \end{pmatrix}$$

$$k(\vec{x}_n,\vec{x}_m) = \vec{\phi}(\vec{x}_n)^{\mathrm{T}}\vec{\phi}(\vec{x}_m) = (0.55 \quad 0.64)\begin{pmatrix} 0.81 \\ 0.27 \end{pmatrix} = 0.628$$



Espace représenté par $\vec{\phi}(\vec{x}) = (\phi_1(\vec{x}), \phi_2(\vec{x}))^{\mathrm{T}}$

Représentation Duale

$$J(\vec{a}) = \vec{a}^{\mathrm{T}} K K \vec{a} - \vec{a}^{\mathrm{T}} K \vec{t} + \vec{t}^{\mathrm{T}} \vec{t} + \lambda \vec{a}^{\mathrm{T}} K \vec{a}$$

On peut démontrer qu'en **fixant à zéro la dérivée** $\nabla J(\vec{a}) = 0$ on obtient

$$\boxed{\vec{a} = (K + \lambda I_N)^{-1} \vec{t}}$$

Vecteur Nx1 contenant les *N* cibles de l'ensemble d'entraînement

Matrice *N* x *N* de Gram

Matrice Identité *N* x *N*

Preuve à faire en devoir

Représentation Duale

Une fois $\vec{a}$ calculée, la **prédiction d'une nouvelle donnée** $\vec{x}$ se fait comme suit

$$\begin{aligned} y(\vec{x}) &= \vec{w}^{\mathrm{T}} \vec{\phi}(\vec{x}) \\ &= \vec{a}^{\mathrm{T}} \Phi \vec{\phi}(\vec{x}) \\ &= \left((K + \lambda I_N)^{-1} \vec{t}\right)^{\mathrm{T}} \Phi \vec{\phi}(\vec{x}) \\ &= \left((K + \lambda I_N)^{-1} \vec{t}\right)^{\mathrm{T}} k(\vec{x}) \end{aligned}$$

où

$$k(\vec{x}) = \left(k(\vec{x}_1, \vec{x}), \ldots, k(\vec{x}_N, \vec{x})\right)^{\mathrm{T}}$$

$$k(\vec{x}_n, \vec{x}) = \vec{\phi}(\vec{x}_n)^{\mathrm{T}} \vec{\phi}(\vec{x})$$

Méthode à noyau duale (version alpha)

Entraînement

Soient les données d'entraînement brutes $D = \{(\vec{x}_1, t_1), \ldots, (\vec{x}_N, t_N)\}$

Mettre les N cibles dans un vecteur à N dimensions $\vec{t}$

Appliquer les fonctions de base à chaque donnée : $\vec{x}_n \rightarrow \vec{\phi}(\vec{x}_n) \quad \forall n$

Calculer la matrice N x N de Gram : $K(\vec{x}_n, \vec{x}_m) = \vec{\phi}(\vec{x}_n)^{\mathrm{T}} \vec{\phi}(\vec{x}_m) \quad \forall n, m$

$\vec{a} = (K + \lambda I_N)^{-1} \vec{t}$

Généralisation

Appliquer les fonctions de base à la donnée à prédire : $\vec{x} \rightarrow \vec{\phi}(\vec{x})$

$\vec{k}(\vec{x})^{\mathrm{T}} = (k(\vec{x}_1, \vec{x}), \ldots, k(\vec{x}_N, \vec{x})) = (\vec{\phi}(\vec{x}_1)^{\mathrm{T}} \vec{\phi}(\vec{x}), \ldots, \vec{\phi}(\vec{x}_N)^{\mathrm{T}} \vec{\phi}(\vec{x}))$

$y_{\vec{a}}(\vec{x}) = \vec{k}(\vec{x})^{\mathrm{T}} \vec{a}$

Peut-on améliorer l’efficacité
des algorithmes de la page précédente?

OUI!

Astuce du noyau (*kernel trick*)

Dans l'algo d'entraînement de la page précédente, on calcule en premier

$$\vec{x}_n \rightarrow \vec{\phi}(\vec{x}_n) \quad \forall n$$

puis ensuite la matrice de Gram

$$K(\vec{x}_n, \vec{x}_m) = \vec{\phi}(\vec{x}_n)^{\mathrm{T}} \vec{\phi}(\vec{x}_m) \quad \forall n, m$$

L'astuce du noyau permet de **calculer K sans avoir à calculer** $\vec{\phi}(\vec{x}_n)$ ce qui est plus efficace

Astuce du noyau (*kernel trick*)

Exemple:

Soit $\vec{x}^{\mathrm{T}} = (x_1, x_2)$ et $\vec{\phi}(\vec{x})^{\mathrm{T}} = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$

Par conséquent

$$\underbrace{k(\vec{x}, \vec{x}') = \vec{\phi}(\vec{x})^{\mathrm{T}} \vec{\phi}(\vec{x}')}_{\text{10 multiplications et 2 additions}}$$

$$= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(x'^2_1, \sqrt{2}x'_1x'_2, x'^2_2)^T$$

$$= x_1^2 x'^2_1 + 2x_1x'_1\, x_2x'_2 + x_2^2 x'^2_2$$

$$= (x_1x'_1 + x_2x'_2)^2$$

$$= \underbrace{(\vec{x}^{\mathrm{T}} \vec{x}')^2}_{\text{3 multiplications et 1 addition}}$$

Astuce du noyau (*kernel trick*)

Une forme générale est le **noyau polynomial**

$$k(\vec{x},\vec{x}')=\left(\vec{x}^{\mathrm{T}}\vec{x}'+c\right)^{M}$$

où c est une constante > 0

On peut montrer que le $\vec{\phi}(\vec{x})$ implicite que ce noyau contient tous les produits possibles entre au plus M éléments de $\vec{x}$

$$M=1 \quad \Rightarrow \quad \vec{\phi}(\vec{x})=\left(c_0, c_0 x_1, \ldots, c_0 x_d\right)$$

Astuce du noyau (*kernel trick*)

Une forme générale est le **noyau polynomial**

$$k(\vec{x},\vec{x}')=\left(\vec{x}^{\mathrm{T}}\vec{x}'+c\right)^{M}$$

Où c est une constante > 0

On peut montrer que le $\vec{\phi}(\vec{x})$ implicite que ce noyau contient tous les produits possibles entre au plus M éléments de $\vec{x}$

$$M=2 \quad \Rightarrow \quad \vec{\phi}(\vec{x})=\big(c_0, c_0 x_1, \ldots, c_d x_d,$$
$$c_{11} x_1^2, c_{12} x_1 x_2, \ldots\big)$$

Astuce du noyau (*kernel trick*)

Une forme générale est le **noyau polynomial**

$$k(\vec{x},\vec{x}')=\left(\vec{x}^{\mathrm{T}}\vec{x}'+c\right)^M$$

Où c est une constante > 0

On peut montrer que le $\vec{\phi}(\vec{x})$ implicite que ce noyau contient tous les produits possibles entre au plus M éléments de $\vec{x}$

$$M=3 \quad \Rightarrow \quad \vec{\phi}(\vec{x})=\left(c_0, c_0x_1, \ldots, c_d x_d,\right.$$
$$c_{11}x_1^2, c_{12}x_1x_2, \ldots,$$
$$\left.c_{111}x_1^3, c_{112}x_1^2x_2, c_{123}, x_1x_2x_3, \ldots\right)$$

Grâce à **l'astuce du noyau,**

on défini un **noyau** $k(\vec{x}_n,\vec{x}_m)$

et non

des **fonctions de base** $\vec{\phi}(\vec{x})$

L'astuce du noyau

Entraînement

Soient les données d'entraînement brutes $D = \{(\vec{x}_1, t_1), \ldots, (\vec{x}_N, t_N)\}$

Mettre les N cibles dans un vecteur à N dimensions $\vec{t}$

Calculer la matrice N x N de Gram : $k(\vec{x}_n, \vec{x}_m) \quad \forall n, m$

$\vec{a} = (K + \lambda I_N)^{-1} \vec{t}$

Généralisation

Calculer le noyau entre chaque donnée d'entrainement et $\vec{x}$: $k(\vec{x})^{\mathrm{T}} = (k(\vec{x}_1, \vec{x}), \ldots, k(\vec{x}_N, \vec{x}))$

$y_{\vec{a}}(\vec{x}) = k(\vec{x})^{\mathrm{T}} \vec{a}$

Quelles noyaux utiliser?

Astuce du noyau (*kernel trick*)

Question : comment peut-on définir des noyaux $k(\vec{x}_n, \vec{x}_m)$ valides
c'est-à-dire des noyaux pour lesquels $k(\vec{x}_n, \vec{x}_m) = \vec{\phi}(\vec{x}_n)^{\mathrm{T}} \vec{\phi}(\vec{x}_m)$

Règles pour **construire des noyaux** valides

$$k(\vec{x}, \vec{x}') = c k_1(\vec{x}, \vec{x}')$$
$$k(\vec{x}, \vec{x}') = k_1(\vec{x}, \vec{x}') + k_2(\vec{x}, \vec{x}')$$
$$k(\vec{x}, \vec{x}') = f(\vec{x}) k_1(\vec{x}, \vec{x}') f(\vec{x}')$$
$$k(\vec{x}, \vec{x}') = q(k_1(\vec{x}, \vec{x}'))$$
$$k(\vec{x}, \vec{x}') = \exp(k_1(\vec{x}, \vec{x}'))$$

$$k(\vec{x}, \vec{x}') = k_1(\vec{x}, \vec{x}') k_2(\vec{x}, \vec{x}')$$
$$k(\vec{x}, \vec{x}') = k_1(\vec{\phi}(\vec{x}), \vec{\phi}(\vec{x}'))$$
$$k(\vec{x}, \vec{x}') = \vec{x}^{\mathrm{T}} \mathrm{A} \vec{x}'$$
$$k(\vec{x}, \vec{x}') = k_1(\vec{x}_a, \vec{x}_a') + k_2(\vec{x}_b, \vec{x}_b')$$
$$k(\vec{x}, \vec{x}') = k_1(\vec{x}_a, \vec{x}_a') k_2(\vec{x}_b, \vec{x}_b')$$

Où c>0, $f(\vec{x})$ est une function, *q(a)* est un polynôme avec coefficients positifs
A est une matrice définie positive et $\vec{x} = (\vec{x}_a, \vec{x}_b)$
les noyaux k_1, k_2 doivent être valides.

Construction de noyaux

Exemple1: prouvons que $ck_1(\vec{x},\vec{x}')$ est valide si c>0 et $k_1(\vec{x},\vec{x}')$ est un noyau valide.

Si $k_1(\vec{x},\vec{x}')$ est un noyau valide alors il existe une fonction de base $\vec{\phi}(\vec{x})$ tel que

$$k_1(\vec{x},\vec{x}') = \vec{\phi}(\vec{x})^{\mathrm{T}} \vec{\phi}(\vec{x}')$$

Par conséquent

$$\begin{aligned} ck_1(\vec{x},\vec{x}') &= c\vec{\phi}(\vec{x})^{\mathrm{T}} \vec{\phi}(\vec{x}') \\ &= \left(\sqrt{c}\vec{\phi}(\vec{x})\right)^{\mathrm{T}} \left(\sqrt{c}\vec{\phi}(\vec{x}')\right) \\ &= \hat{\phi}(\vec{x})^{\mathrm{T}} \hat{\phi}(\vec{x}) \end{aligned}$$

CQFD ■

Construction de noyaux

Exemple2: prouvons que le noyau gaussien est valide: $k(\vec{x},\vec{x}') = e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}}$

Considérant que $\|\vec{x}-\vec{x}'\|^2 = \vec{x}^{\mathrm{T}}\vec{x} + \vec{x}'^{\mathrm{T}}\,\vec{x}' - 2\vec{x}^{\mathrm{T}}\vec{x}'$

On obtient que

$$\begin{aligned} k(\vec{x},\vec{x}') &= e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}} \\ &= e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}+\vec{x}'^{\mathrm{T}}\,\vec{x}'-2\vec{x}^{\mathrm{T}}\vec{x}'}{2\sigma^2}} \\ &= e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}}{2\sigma^2}} e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}'}{2\sigma^2}} e^{\frac{\vec{x}'^{\mathrm{T}}\,\vec{x}'}{\sigma^2}} \end{aligned}$$

Construction de noyaux

Exemple2: prouvons que le noyau gaussien est valide: $k(\vec{x},\vec{x}')=e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}}$

$\vec{x}^{\mathrm{T}}\vec{x}'=k_1(\vec{x},\vec{x}')\Rightarrow valide$

$$k(\vec{x},\vec{x}')=e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}}{2\sigma^2}}e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}'}{2\sigma^2}}e^{\frac{\vec{x}'^{\mathrm{T}}\vec{x}'}{\sigma^2}}$$

Construction de noyaux

Exemple2: prouvons que le noyau gaussien est valide: $k(\vec{x},\vec{x}')=e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}}$

$\vec{x}^{\mathrm{T}}\vec{x}'=k_1(\vec{x},\vec{x}')\Rightarrow valide$

$k_2(\vec{x},\vec{x}')=ck_1(\vec{x},\vec{x}')$

$$k(\vec{x},\vec{x}')=e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}}{2\sigma^2}}e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}'}{2\sigma^2}}e^{-\frac{\vec{x}'^{\mathrm{T}}\vec{x}'}{2\sigma^2}}$$

Construction de noyaux

Exemple2: prouvons que le noyau gaussien est valide: $k(\vec{x},\vec{x}')=e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}}$

$$\vec{x}^{\mathrm{T}}\vec{x}'=k_1(\vec{x},\vec{x}')\Rightarrow valide$$
$$k_2(\vec{x},\vec{x}')=ck_1(\vec{x},\vec{x}')$$
$$k_3(\vec{x},\vec{x}')=\exp(k_2(\vec{x},\vec{x}'))$$

$$k(\vec{x},\vec{x}')=e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}}{2\sigma^2}}e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}'}{2\sigma^2}}e^{-\frac{\vec{x}'^{\mathrm{T}}\vec{x}'}{2\sigma^2}}$$

Construction de noyaux

Exemple2: prouvons que le noyau gaussien est valide: $k(\vec{x},\vec{x}')=e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}}$

$$\vec{x}^{\mathrm{T}}\vec{x}'=k_1(\vec{x},\vec{x}')\Rightarrow valide$$
$$k_2(\vec{x},\vec{x}')=ck_1(\vec{x},\vec{x}')$$
$$k_3(\vec{x},\vec{x}')=\exp(k_2(\vec{x},\vec{x}'))$$
$$k(\vec{x},\vec{x}')=f(\vec{x})k_3(\vec{x},\vec{x}')f(\vec{x}')$$

$$k(\vec{x},\vec{x}')=e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}}{2\sigma^2}}e^{-\frac{\vec{x}^{\mathrm{T}}\vec{x}'}{2\sigma^2}}e^{-\frac{\vec{x}'^{\mathrm{T}}\vec{x}'}{2\sigma^2}}$$

CQFD ■

Capacité d'un noyau

- **Noyau polynomial:** $k(\vec{x}, \vec{x}') = \left(\vec{x}^{\mathrm{T}} \vec{x}' + c\right)^M$
  - ➢ plus M **est grand**, plus le modèle a de la capacité
- **Noyau gaussien:** $k(\vec{x}, \vec{x}') = e^{-\frac{\|\vec{x}-\vec{x}'\|^2}{2\sigma^2}}$
  - ➢ plus σ^2 **est petit**, plus le modèle a de la capacité

Résumé

- **Problème:** $\vec{w} = \arg\min_{\vec{w}} \sum_{n=1}^{N} \left(\vec{w}^T \vec{\phi}(\vec{x}_n) - t_n\right)^2 + \lambda \vec{w}^T \vec{w}$
- **Paramètres** : $\vec{w} = \sum_{n=1}^{N} a_n \vec{\phi}(\vec{x}_n) = \Phi^{\mathrm{T}} \vec{a}$ somme pondérée des entrées
- **Entraînement** : $\vec{a} = (K + \lambda I_N)^{-1} \vec{t}$
  Matrice de Gram
- **Prédiction**: $y(\vec{x}) = \sum_{n=1}^{N} k(\vec{x}, \vec{x}_n) a_n$
  Noyau
- **Hyper-paramètres**:
  - λ
  - c et M pour le noyau polynomial
  - σ pour le noyau gaussien