

Introducción a la Toma de Decisiones Basadas en Datos

Teodoro Alamo Cantarero

**Grupo de Estimación, Predicción y Control
Escuela Superior de Ingenieros
Universidad de Sevilla**

21 abril 2017

Departamento Informática y Automática. UNED.

- Pilares fundamentales de esta disciplina.
- Fuentes bibliográficas y recursos on-line.
- Reseña histórica del método de los mínimos cuadrados.
- Limitaciones de los mínimos cuadrados.
- Vías para la generalización robusta.
- Metodologías de Regularización.
- Métodos Kernel.

- **Álgebra:**

- Matrices definidas positivas.
- Descomposición Cholesky.
- Lema de inversión.

- **Estadística:**

- Covarianza, Gaussianas, etc.
- Análisis Bayesiano, Teoría de la información.
- Métodos Kernel.

- **Optimización:**

- Convexidad.
- Minimización funciones no diferenciables.
- Dualidad.
- FISTA.

- **Informática:**

- Bases de datos (SQL y No-SQL).
- Python, R, Matlab, etc.

- **Machine Learning** (Andrew Ng, Stanford):
<https://www.coursera.org/learn/machine-learning>
- **Introduction to Machine Learning** (John Paisley, Columbia):
<https://www.edx.org/course/machine-learning-columbiacsmm-102x>
- **Clases magistrales:**
<http://videlectures.net/>
- **Machine Learning Summer Schools:**
<http://mlss.cc/>

-  S. Marsland. *Machine Learning. An Algorithmic Perspective*. CRC Press, Taylor & Francis. 2009.
-  C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer. 2006.
-  K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press. 2012.
-  I. Goodfellow et al. *Deep Learning*. The MIT Press. 2016.
-  B. Schölkopf y A.J. Smola. *Learning with Kernels*. The MIT Press. 2002.
-  C.E. Rasmussen y C.K.I. Williams *Gaussian Processes for Machine Learning*. The MIT Press. 2006.

Ejemplo motivador

La regla de Titius-Bode, desarrollada a finales del siglo XVIII servía para predecir la existencia y localización de los planetas en nuestro sistema solar.

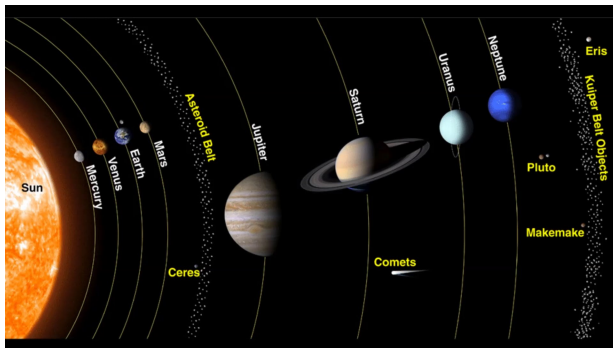


Figure: Sistema solar

Regla de los astrónomos Titius y Bode (1798)

Si uno ordena los planetas con respecto a la distancia al Sol, la distancia d_n en unidades astronómicas del planeta en la posición n se puede aproximar por

$$d_n \approx 0.4 + 0.3 \cdot 2^m,$$

donde m es $-\infty$ para Mercurio, 0 para Venus, 1 para la Tierra, 2 para Marte, etc.

n	m	Planeta	Distancia (UA)	Estimación (UA)
1	$-\infty$	Mercurio	0.39	0.4
2	0	Venus	0.72	0.7
3	1	Tierra	1.00	1.0
4	2	Marte	1.52	1.6
6	4	Júpiter	5.20	5.2
7	5	Saturno	9.55	10.0

Regla de los astrónomos Titius y Bode

- La regla funcionaba razonablemente bien para los cinco planetas conocidos en 1798.
- Tres años después de publicar la regla, se descubrió Urano a una distancia de 19.22 UA.
- Urano ocupaba la octava posición ($n = 8$, $m = 6$) y la regla da una estimación de 19.6 UA.
- La comunidad científica de la época quedó convencida de la validez de la regla.
- En la regla m salta de 2 para Marte a 4 para Júpiter.
- Este salto se podría explicar con la existencia de un planeta entre Marte y Júpiter ($m = 3$).

- El 1 de enero de 1801 el astrónomo italiano Joseph Piazzi descubrió el planetoide Ceres.
- Ceres estaba a una distancia de 2.77 UA.
- Si asumimos que Ceres es el quinto planeta ($n = 5$ and $m = 3$), la regla de Titius-Bode da una distancia de 2.8 UA.
- Piazzi pudo observar Ceres sólo 41 días porque su órbita se perdió detrás del sol.
- El reducido conjunto de observaciones de Ceres se publicaron en junio de 1801.
- El objetivo de la comunidad científica fue predecir, con estos datos, la posición de la siguiente aparición visible de Ceres.

Gauss y Ceres

- En septiembre de 1801, distintas predicciones sobre la futura reaparición de Ceres fueron publicadas.
- Entre ellas estaba la de un joven de 24 años llamado Carl Friedrich Gauss.
- Cuando Ceres fue redescubierto el 7 de diciembre de 1801, se encontró casi exactamente donde Gauss había predicho.
- Gauss había utilizado una metodología, creada por él, que luego vino a llamarse método de los mínimos cuadrados.



Método de los mínimos cuadrados: ejemplo

Usaremos el método de los mínimos cuadrados para obtener una regla similar a la de Titius-Bode.

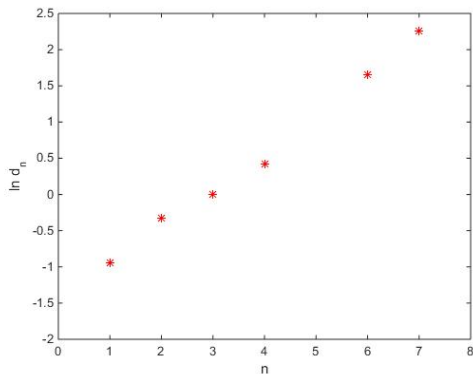


Figure: Logaritmo de la distancia al sol de Mercurio ($n=1$), Venus ($n=2$), Tierra ($n=3$), Marte ($n=4$), Júpiter ($n=6$) y Saturno ($n=7$).

Método de los mínimos cuadrados: ejemplo

El logaritmo de la distancia d_n se puede aproximar razonablemente con una recta de la forma:

$$\ln d_n \approx an + b,$$

donde n representa la posición que del planeta en cuestión mientras que a y b son parámetros que deben ajustarse con los datos experimentales. En forma matricial:

$$y_{\mathcal{D}} = \begin{bmatrix} \ln d_1 \\ \ln d_2 \\ \ln d_3 \\ \ln d_4 \\ \ln d_6 \\ \ln d_7 \end{bmatrix} \approx \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 6 & 1 \\ 7 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = R \begin{bmatrix} a \\ b \end{bmatrix} = R\theta.$$

Método de los mínimos cuadrados: ejemplo

Tenemos dos parámetros (a y b) y seis observaciones (el vector $y_{\mathcal{D}}$ tiene 6 componentes). Asumimos que existe un vector de errores e que hace que la aproximación ($y_{\mathcal{D}} \approx R\theta$) se convierta en una igualdad. Esto es,

$$y_{\mathcal{D}} = R\theta + e.$$

Si ahora premultiplicamos por la traspuesta de R ambos términos de la igualdad obtenemos:

$$R^{\top} y_{\mathcal{D}} = R^{\top} R\theta + R^{\top} e.$$

Despreciando el término de error $R^{\top} e$:

$$R^{\top} y_{\mathcal{D}} \approx R^{\top} R\theta.$$

$$\theta \approx (R^{\top} R)^{-1} R^{\top} y_{\mathcal{D}}.$$

Método de los mínimos cuadrados: ejemplo

Si aplicamos esta metodología a nuestro ejemplo obtenemos:

$$\theta = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0.5239 \\ -1.4993 \end{bmatrix}.$$

De esta forma hemos obtenido una regla similar a la propuesta por Titius-Bode: $\ln d_n \approx 0.5239n - 1.4993$.

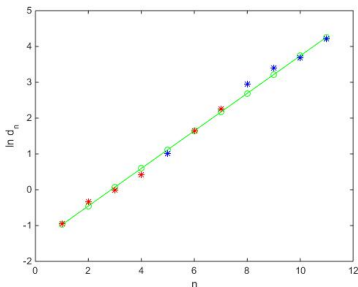


Figure: En azul los planetas descubiertos después de 1780: Ceres $n = 5$, Urano $n = 8$, Neptuno $n = 9$, Pluto $n = 10$ y Eris $n = 11$.

Aunque la regla de Titius-Bode funciona razonablemente bien para una serie de planetas:

- Los astrónomos no han conseguido detectar un patrón similar en otros sistemas solares.
- Tampoco se ha conseguido justificar dicha regla utilizando la mecánica clásica.
- La regla es algo arbitraria porque depende de cómo ordenamos los planetas (algunos no se consideran por ser demasiado pequeños, otros se agrupan por cercanía).

Actualmente se piensa que la regla se debe tan sólo a una casualidad numérica que sirvió para descubrir Ceres y animar a Gauss a publicar la primera utilización del método de los mínimos cuadrados.

Supongamos que se dispone del conjunto de datos

$$(y_i, x_i), \quad i = 1, \dots, N.$$

Asumamos que existe un vector paramétrico $\theta \in \mathbb{R}^{n_\theta}$ y una función $\varphi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_\theta}$ tal que

$$y_i = \varphi(x_i)^\top \theta + e_i, \quad i = 1, \dots, N$$

donde e_i representa un término de error.

Los mínimos cuadrados obtienen el vector paramétrico de la resolución del siguiente problema de optimización:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{2} \sum_{i=1}^N (y_i - \varphi(x_i)^\top \theta)^2.$$

Limitaciones de los mínimos cuadrados

- En algunas ocasiones, el problema no está bien condicionado numéricamente.
- No es sencillo incorporar el conocimiento previo que se pueda tener sobre θ .
- Si el número de datos es reducido, no es razonable utilizar regresores $\varphi(\cdot)$ de gran dimensión (sobreparametrización).
- Nos vemos obligados a proponer explícitamente el regresor $\varphi(\cdot)$.

Decimos que nuestra metodología generaliza de forma robusta si la estimación $y \approx \varphi(x)^\top \hat{\theta}$ funciona bien no sólo en el set de entrenamiento

$$(y_i, x_i), \quad i = 1, \dots, N,$$

sino también en un set de validación

$$(y_i, x_i), \quad i = N + 1, \dots, N + M.$$

Las tres vías para incrementar la robustez son:

- 1 Incrementar el número y representabilidad de los datos de entrenamiento.
- 2 Reducir el número de parámetros a ajustar (dimensión de θ).
- 3 Reducir la norma (tamaño) del vector $\theta \Rightarrow$ regularización.

- Regularización Tikhonov (Ridge regression):

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{2} \theta^\top \Sigma_\theta^{-1} \theta + \frac{\tau}{2} \sum_{i=1}^N (y_i - \varphi(x_i)^\top \theta)^2.$$

- Lasso:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \lambda \|\theta\|_1 + \frac{\tau}{2} \sum_{i=1}^N (y_i - \varphi(x_i)^\top \theta)^2.$$

- Elastic Net:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{2} \theta^\top \Sigma_\theta^{-1} \theta + \lambda \|\theta\|_1 + \frac{\tau}{2} \sum_{i=1}^N (y_i - \varphi(x_i)^\top \theta)^2.$$

El parámetro $\tau > 0$ nos permite llegar a un compromiso entre el error cometido en el set de entrenamiento y el tamaño de $\hat{\theta}$.

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{2} \theta^\top \Sigma_\theta^{-1} \theta + \frac{\tau}{2} \sum_{i=1}^N (y_i - \varphi(x_i)^\top \theta)^2.$$

Si denotamos

$$R = \begin{bmatrix} \varphi^\top(x_1) \\ \varphi^\top(x_2) \\ \vdots \\ \varphi^\top(x_N) \end{bmatrix}, \quad y_{\mathcal{D}} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}.$$

la estimación Ridge Regression tiene una solución explícita:

$$\hat{\theta} = (R^\top R + \frac{1}{\tau} \Sigma_\theta^{-1})^{-1} R^\top y_{\mathcal{D}}.$$

Es conveniente “centrar” los datos para que $\theta = 0$ tenga un error de media nula en el set de entrenamiento.

Regularizaciones que inducen soluciones de baja cardinalidad

Consideremos la regularización Lasso:

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^{n_\theta}} \lambda \|\theta\|_1 + \frac{\tau}{2} \sum_{i=1}^N (y_i - \varphi(x_i)^\top \theta)^2.$$

- En este caso no tenemos solución explícita y tenemos que recurrir a optimización numérica (por ejemplo FISTA).
- El estimador Lasso se encarga, de forma “automatizada”, de seleccionar qué componentes del regresor son relevantes y cuáles no.
- La fracción de componentes de θ que son igual a cero es alta si elegimos λ grande.
- Se pueden obtener estimadores muy robustos a pesar de que tengamos pocos datos de partida y un regresor $\varphi(\cdot)$ de alta dimensión.

Sistemas variantes en el tiempo

- Supongamos que y_k representa la salida de un sistema muestreado que cambia con el tiempo.
- El objetivo es predecir y_{k+1} a partir del regresor φ_{k+1} con el conocimiento que tenemos hasta la muestra k .
- Definimos el funcional:

$$J_k(\theta) = \frac{1}{2}\theta^\top \Sigma_\theta^{-1}\theta + \frac{\tau}{2} \sum_{i=1}^k \alpha^{k-i} (y_i - \varphi_i^\top \theta)^2,$$

donde $\alpha \in (0, 1]$ representa un factor de olvido que hace que le demos más relevancia a los errores recientes.

- En Ridge Regression, obtenemos la estimación como $y_{k+1} \approx \varphi(x_{k+1})^\top \tilde{\theta}_{k+1}$ donde

$$\tilde{\theta}_{k+1} = \arg \min J_k(\theta).$$

- Implementación factor olvido:

- $\tilde{H}_k = \frac{1}{\alpha} \tilde{P}_k.$

- Corrección:

- $K_k = \frac{\tau \tilde{H}_k \varphi_k}{1 + \tau \varphi_k^\top \tilde{H}_k \varphi_k},$

- $\beta_k = \tilde{\theta}_k + K_k (y_k - \varphi_k^\top \tilde{\theta}_k),$

- $\hat{H}_k = (I - K_k \varphi_k^\top) \tilde{H}_k.$

- Regularización:

- $G_k = \Sigma_\theta (\Sigma_\theta + (1 - \alpha) \hat{H}_k)^{-1},$

- $\hat{P}_k = G_k \hat{H}_k,$

- $\hat{\theta}_k = G_k \beta_k.$

- Predicción:

- $\tilde{P}_{k+1} = \hat{P}_k,$

- $\tilde{\theta}_{k+1} = \hat{\theta}_k.$

Generalizaciones y filtro de Kalman

- La estructura de la implementación recursiva es similar al filtro de Kalman.
- El factor de olvido se implementa como: $\tilde{H}_k = \frac{1}{\alpha} \tilde{P}_k$.
- Se podría implementar un factor de olvido no homogéneo como:

$$\tilde{H}_k = D_\alpha \tilde{P}_k D_\alpha,$$

donde

$$D = \begin{bmatrix} \frac{1}{\sqrt{\alpha_1}} & & & & \\ & \frac{1}{\sqrt{\alpha_2}} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \frac{1}{\sqrt{\alpha_{n_\theta}}} \end{bmatrix}.$$

- También se podría añadir una matriz de covarianza a \tilde{P}_{k+1} , como se realiza en el filtro de Kalman para modelar una deriva en los parámetros del sistema.

Detección en línea de fuga en red distribución

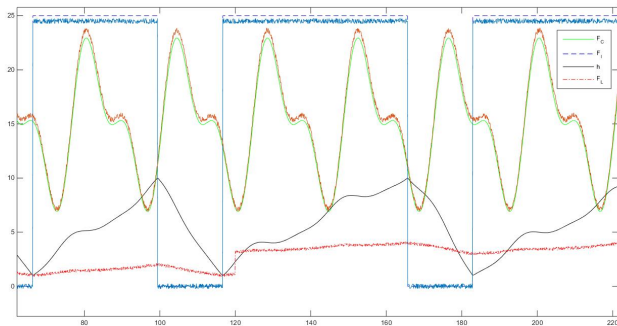


Figure: F_C : Caudal suministrado a la población, F_I : Caudal entrada al depósito, h : Nivel depósito, F_L : Pérdidas físicas (fugas).

El objetivo de la identificación es estimar en línea las pérdidas a baja y alta presión así como los factores de calibración c_1 y c_2 asociados a los caudalímetros.

Detección en línea de fuga en red distribución

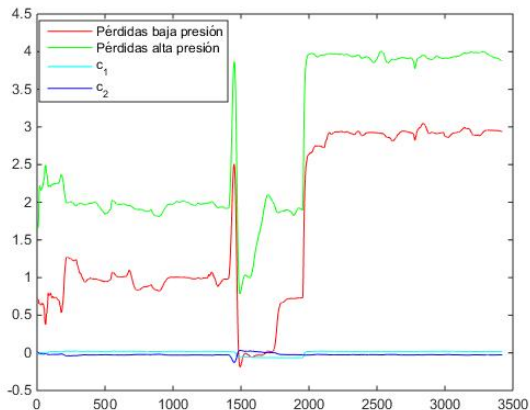


Figure: Factor olvido homogéneo : las pérdidas se hacen negativas y se compensan con los factores de calibración instrumental c_1 y c_2 .

Detección en línea de fuga en red distribución

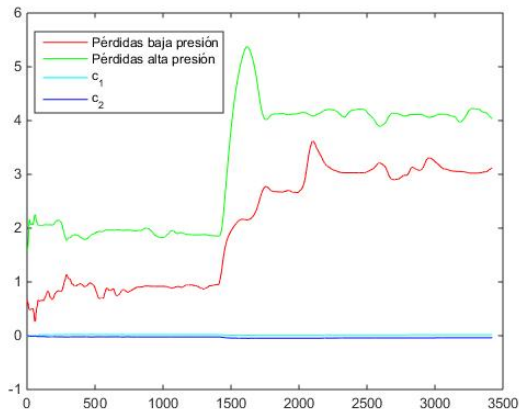


Figure: Factor olvido no homogéneo: Al asignar mayor factor de olvido a las pérdidas, la nueva fuga se identifica de forma satisfactoria y no se compensa tanto con los factores de calibración.

El estimador que nos proporciona Ridge Regression es de la forma

$$y \approx \varphi(x)^\top \hat{\theta}.$$

- La elección de la función $\varphi(\cdot)$ juega un papel determinante.
- La forma tradicional de proceder es probar diferentes posibilidades para $\varphi(\cdot)$ y ver cuál proporciona mejor resultado en un set de validación.
- Esto puede ser muy engorroso, sobre todo si x tiene una dimensión alta y queremos introducir relaciones no lineales.
- Considerar relaciones polinómicas de grado d entre todas las componentes de x nos proporciona una dimensión de θ muy alta para valores moderados de d si la dimensión de x no es baja.

- En Ridge Regression, la estimación para un nuevo vector x depende **tan sólo** de los productos escalares

$$\begin{aligned}\varphi(x)^\top \Sigma_\theta \varphi(x_i), & \quad i = 1, \dots, N. \\ \varphi(x_i)^\top \Sigma_\theta \varphi(x_j), & \quad i = 1, \dots, N \quad j = 1, \dots, N.\end{aligned}$$

- Como demostraremos, no es necesario conocer de forma explícita la función $\varphi(\cdot)$.
- Tan sólo** es necesario conocer cómo calcular $\varphi(x_a)^\top \Sigma_\theta \varphi(x_b)$ para cualquier par x_a y x_b .
- Esto se realiza definiendo una función Kernel:

$$k(x_a, x_b) = \varphi^\top(x_a) \Sigma_\theta \varphi(x_b).$$

- For ejemplo, podríamos tomar $k(x_a, x_b) = \exp\left(-\frac{\|x_a - x_b\|^2}{\sigma^2}\right)$ donde σ es un parámetro a ajustar.

Para demostrar que la estimación de la Ridge regresión se puede expresar en términos de funciones kernel nos basaremos en el conocido lema de inversión:

Lema de Inversión

Supongamos que $\Sigma_a > 0$. Entonces:

- $\Sigma_a^{-1} + R^T R$ y $I + R \Sigma_a R^T$ son invertibles.
- $(\Sigma_a^{-1} + R^T R)^{-1} = \Sigma_a - \Sigma_a R^T (I + R \Sigma_a R^T)^{-1} R \Sigma_a$.
- $(\Sigma_a^{-1} + R^T R)^{-1} R^T = \Sigma_a R^T (I + R \Sigma_a R^T)^{-1}$.

De la aplicación directa del tercer punto del lema de inversión obtenemos:

$$\begin{aligned}\hat{\theta} &= (R^T R + \frac{1}{\tau} \Sigma_{\theta}^{-1})^{-1} R^T y_D \\ &= \tau \Sigma_{\theta} R^T (I + \tau R \Sigma_{\theta} R^T)^{-1} y_D \\ &= \Sigma_{\theta} R^T (\frac{1}{\tau} I + R \Sigma_{\theta} R^T)^{-1} y_D.\end{aligned}$$

Por tanto:

$$\hat{y} = \varphi(x)^T \hat{\theta} = \varphi^T(x) \Sigma_{\theta} R^T (\frac{1}{\tau} I + R \Sigma_{\theta} R^T)^{-1} y_D.$$

Por otro lado,

$$\begin{aligned}\varphi^T(x) \Sigma_{\theta} R^T &= \varphi^T(x) \Sigma_{\theta} [\varphi(x_1) \quad \varphi(x_2) \quad \dots \quad \varphi(x_N)] \\ &= [k(x, x_1) \quad k(x, x_2) \quad \dots \quad k(x, x_N)].\end{aligned}$$

Comprobamos que la matriz $K = R\Sigma_{\theta}R^{\top}$ también se puede escribir en términos de $k(\cdot, \cdot)$:

$$\begin{aligned} K &= \begin{bmatrix} \varphi^{\top}(x_1) \\ \varphi^{\top}(x_2) \\ \vdots \\ \varphi^{\top}(x_N) \end{bmatrix} \Sigma_{\theta} \begin{bmatrix} \varphi(x_1) & \varphi(x_2) & \dots & \varphi(x_N) \end{bmatrix} \\ &= \begin{bmatrix} \varphi^{\top}(x_1)\Sigma_{\theta}\varphi(x_1) & \varphi^{\top}(x_1)\Sigma_{\theta}\varphi(x_2) & \dots & \varphi^{\top}(x_1)\Sigma_{\theta}\varphi(x_N) \\ \varphi^{\top}(x_2)\Sigma_{\theta}\varphi(x_1) & \varphi^{\top}(x_2)\Sigma_{\theta}\varphi(x_2) & \dots & \varphi^{\top}(x_2)\Sigma_{\theta}\varphi(x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi^{\top}(x_N)\Sigma_{\theta}\varphi(x_1) & \varphi^{\top}(x_N)\Sigma_{\theta}\varphi(x_2) & \dots & \varphi^{\top}(x_N)\Sigma_{\theta}\varphi(x_N) \end{bmatrix} \\ &= \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix}. \end{aligned}$$

La estimación \hat{y} para x es

$$\hat{y} = \begin{bmatrix} k(x, x_1) \\ k(x, x_2) \\ \vdots \\ k(x, x_N) \end{bmatrix}^T \left(\frac{1}{\tau} \mathbf{I} + \mathbf{K} \right)^{-1} y_{\mathcal{D}},$$

donde

$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix}.$$

- La estimación \hat{y} se puede obtener directamente de la función kernel $k(\cdot, \cdot)$ y el vector $y_{\mathcal{D}}$.
- La función Kernel no se puede elegir de forma arbitraria porque $\mathbf{K} = \mathbf{R}^T \Sigma_{\theta} \mathbf{R}$ debe ser semidefinida positiva.

Kernels más populares

Las funciones Kernel más populares en el contexto de Machine Learning son:

- Lineal: $K(x_a, x_b) = x_a^\top x_b$.
- Polinomial: $K(x_a, x_b) = (c + x_a^\top x_b)^d$.
- Bases radiales : $K(x_a, x_b) = \exp\left(\frac{-\|x_a - x_b\|^2}{\sigma^2}\right)$.
- Sigmoidal: $K(x_a, x_b) = \tanh(c_a x_a^\top x_b + c_b)$.

Excepto para el kernel lineal, hay algunos parámetros de ajuste (hiper-parámetros) que se pueden determinar utilizando distintos enfoques:

- Maximización de la verosimilitud.
- Validación cruzada.
- Evaluación del rendimiento en un set de validación.
- Etc.

Ejemplo: regresión polinómica

- Supongamos que la variable y depende polinómicamente de $x = [z_1 \ z_2 \ z_3]^T$.
- Si asumimos una dependencia cúbica, el regresor $\varphi(x)$ debería contener los siguientes 20 términos

$$1, z_1, z_2, z_3, z_1^2, z_1 z_2, z_1 z_3, z_2^2, z_2 z_3, z_3^2, \\ z_1 z_2 z_3, z_1^3, z_1^2 z_2, z_1^2 z_3, z_2^3, z_2^2 z_1, z_2^2 z_3, z_3^3, z_3^2 z_1, z_3^2 z_2.$$

- Utilizar este regresor es equivalente a tomar como kernel:

$$k(x_a, x_b) = (1 + x_a^T x_b)^3.$$

- La dimensión del regresor en un contexto polinomial crece excesivamente con el grado d y la dimensión de x .
- Utilizando el kernel $(1 + x_a^T x_b)^d$ se evitan estos problemas (aunque las matrices implicadas en la regresión crecen linealmente con el número de datos).

En esta presentación se han abordado los siguientes aspectos de la toma de decisiones basadas en datos:

- Pilares fundamentales de esta disciplina.
- Fuentes bibliográficas y recursos on-line.
- Reseña histórica del método de los mínimos cuadrados.
- Limitaciones de los mínimos cuadrados.
- Vías para la generalización robusta.
- Metodologías de regularización.
- Métodos Kernel.