Clase 21 - Procesos Gaussianos II

Aprendizaje de Máquinas - MA5204

Felipe Tobar

Department of Mathematical Engineering &
Center for Mathematical Modelling
Universidad de Chile

15 de junio de 2021

UNIVERSIDAD
DE CHILE

¿Qué es entrenar un $\mathcal{GP}$?

Vimos que dada una función de covarianza podemos representar el proceso, y podemos encontrar analíticamente la densidad posterior de nuestra función $f(\cdot)$ condicionando a las observaciones. Pero la forma que tendrá la posterior y la función fuera de las observaciones dependerá fuertemente en nuestra función kernel escogida, en este sentido, para un kernel dado nos gustaría encontrar los parámetros de este que mejor representen nuestra función a estimar.

Nos referiremos a entrenar u optimizar un $\mathcal{GP}$ cuando queremos obtener los hyperparámetros, es decir los parámetros del kernel (los denotamos θ) y la varianza del ruido (la denotamos σ_n^2) si es que aplica.

Procesos Gaussianos - Entrenamiento

Consideremos la *verosimilitud marginal*, obtenida marginalizando sobre la función $f(\cdot)$, donde dado un conjunto de entrenamiento $(X, Y) = \{(x_i, y_i)\}_{i=1}^n$, esta dada por

$$\begin{aligned}\mathbb{P}(Y|X,\theta,\sigma) &= \int \mathbb{P}(Y|f,X,\theta,\sigma_n)p(f|X,\theta,\sigma)df \\ &= \frac{1}{(2\pi|\mathbf{K}_y|)^{\frac{n}{2}}}\exp\left(-\frac{1}{2}(Y-\mathbf{m})^T\mathbf{K}_y^{-1}(Y-\mathbf{m})\right)\end{aligned}$$

Donde $\mathbf{m} = m(X)$ y $\mathbf{K}_y = K_\theta(X,X) + \sigma_n^2\mathbb{I}$, la matriz de covarianza dados los parámetros θ agregando el término de la diagonal correspondiente al ruido. Considerando la log-verosimilitud negativa (NLL)

$$NLL = -\log\mathbb{P}(Y|X,\theta,\sigma_n)$$

$$NLL = \underbrace{\frac{1}{2}\log|\mathbf{K_y}|}_{\substack{\text{Penalización}\\ \text{por}\\ \text{complejidad}}} + \underbrace{\frac{1}{2}(Y-\mathbf{m})^T\mathbf{K}_y^{-1}(Y-\mathbf{m})}_{\text{Data fit (Única parte que depende de } Y)} + \underbrace{\frac{n}{2}\log 2\pi}_{\substack{\text{Constante de}\\ \text{normalización}}}$$

Procesos Gaussianos - Entrenamiento

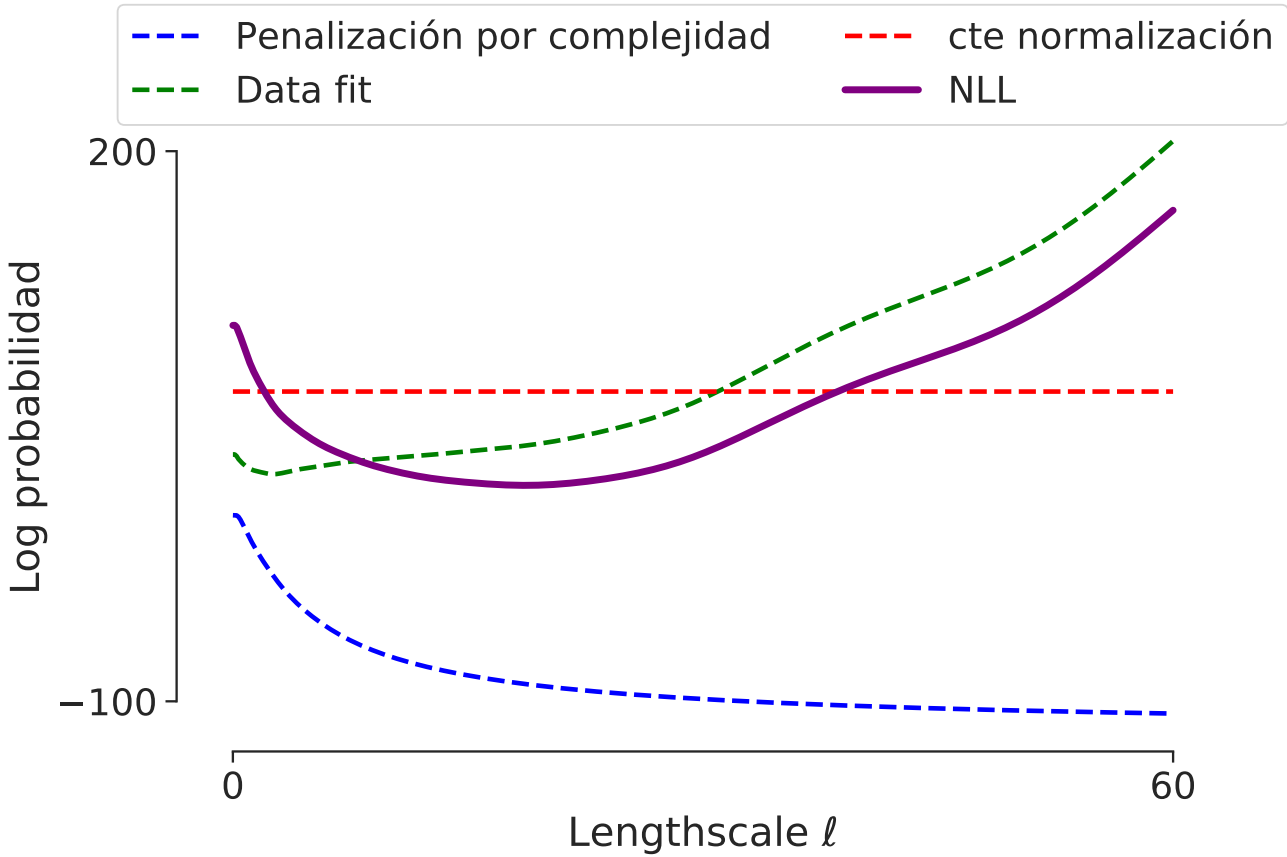

Fig.. Log verosimilitud marginal negativa (NLL) en función del *lengthscale* (ℓ) para señal sintetica, se mantienen constantes los otros parámetros del $\mathcal{GP}$.

Procesos Gaussianos - Optimización

Como contamos con una expresión cerrada para la NLL, podemos utilizar métodos clásicos de optimización, utilizando L-BFGS en el ejemplo anterior

	σ_{ruido}	ℓ	$\sigma_{\text{señal}}$	NLL
Sin entrenar	0.2	3.1622	1	**55.3538**
Entrenado	0.2067	18.7267	0.9956	**17.6945**

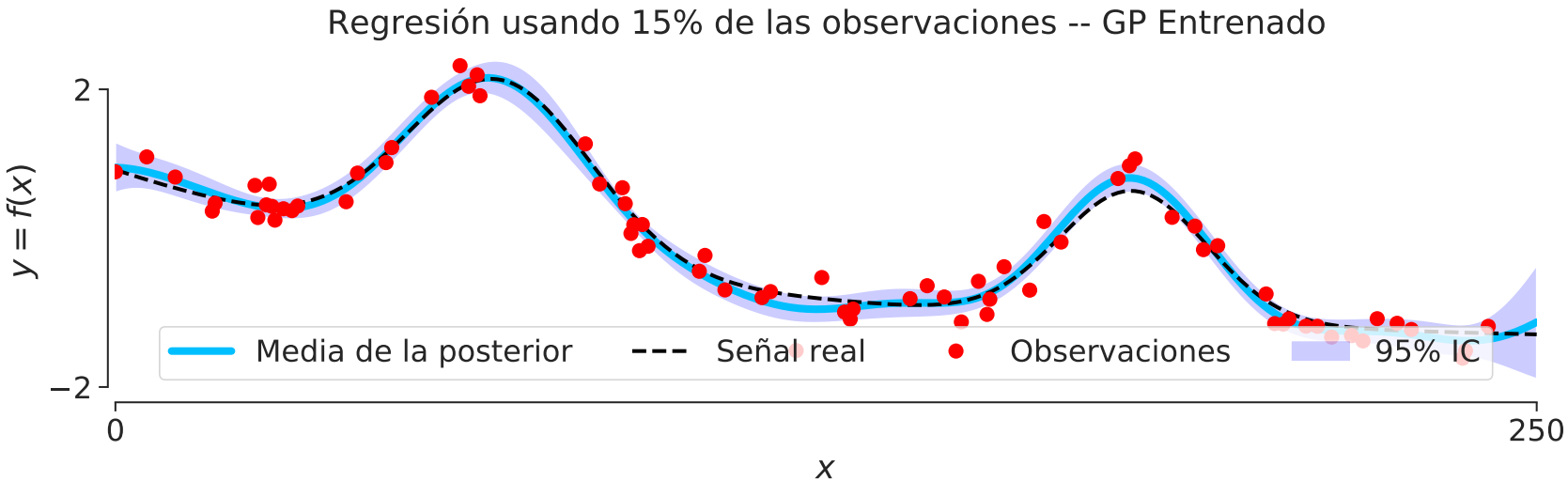

Fig.. Regresión con $\mathcal{GP}$ para señal sintetica usando el 15 % de los datos muestreados de forma no uniforme y contaminados con ruido Gaussiano, utilizando un $\mathcal{GP}$ de media nula y kernel SE; Modelo optimizado utilizando L-BFGS.

Función de covarianza (kernels)

Hasta el momento solo hemos utilizado un tipo de función de covarianza, el llamado kernel *Squared Exponential* (SE), conocido también como kernel RBF. En esta sección mostraremos distintos tipos de funciones de covarianza y los distintos tipos de funciones que generan.

$$K_{SE}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

Es importante denotar tipos de familias de funciones de covarianza, si la función solo depende de la diferencia, es decir $k(x, x') = k(x - x')$ se le llamará *kernel estacionario*, más aún, si depende solo de la norma de la diferencia $k(x, x') = k(|x - x'|)$ se le llamará *kernel isométrico*, un ejemplo de esto es el kernel SE.

Es de notar que kernels estacionarios hacen que la covarianza entre puntos sea invariante a traslaciones en el espacio de entradas. Una noción importante es que un kernel puede ser visto como una medida de similaridad entre puntos, y en el caso de kernels estacionarios, mientras más cercanos estén dos puntos más símiles serán.
A continuación se presentan algunos tipos de kernel:

Rational Quadratic (RQ)

Este kernel viene dado por la siguiente expresión

$$K_{RQ}(x, x') = \sigma^2 \left(1 + \frac{(x - x')^2}{2\alpha\ell^2}\right)^{-\alpha}$$

donde α es un parámetro de variación de escala, notar que cuando $\alpha \to \infty$ el kernel tiende a uno SE.

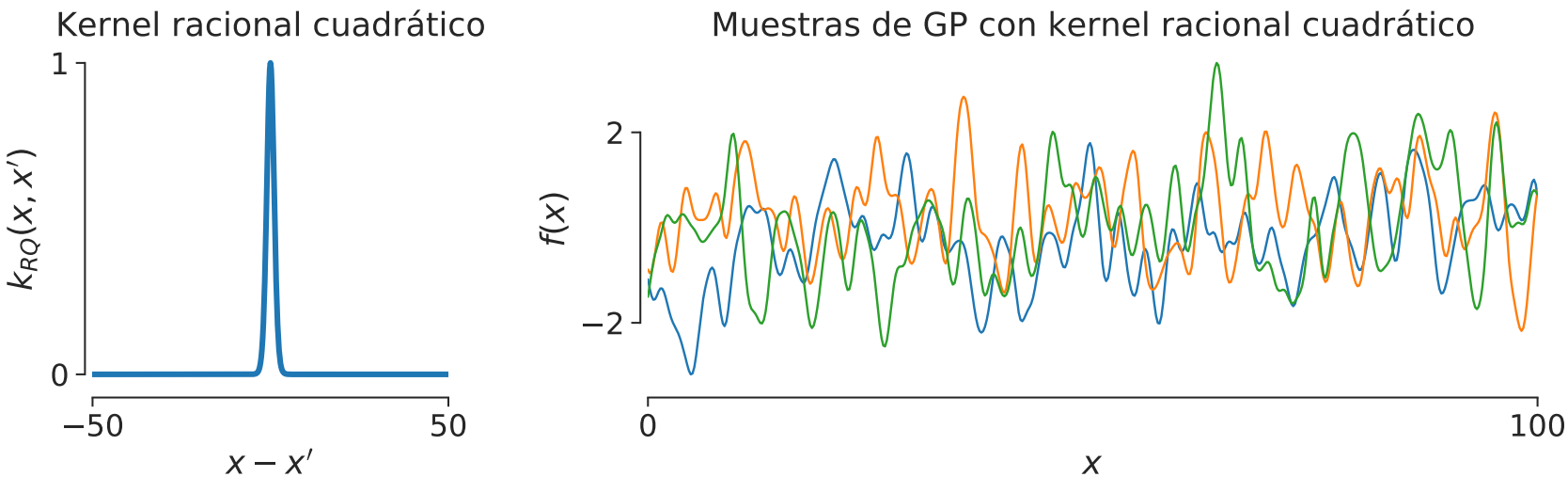

Fig.. Kernel *Rational Quadratic*, en la izquierda se muestra la covarianza en función de su argumento $\tau = x - x'$, a la derecha de un $\mathcal{GP}$ usando un kernel RQ.

Kernel Periódico y localmente periódico

Descrito por las ecuaciones:

$$K_P(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{\ell^2}\right)$$

$$K_{LP}(x, x') = \sigma^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right) \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{\ell^2}\right)$$

Donde el parámetro p controla el periodo de la función

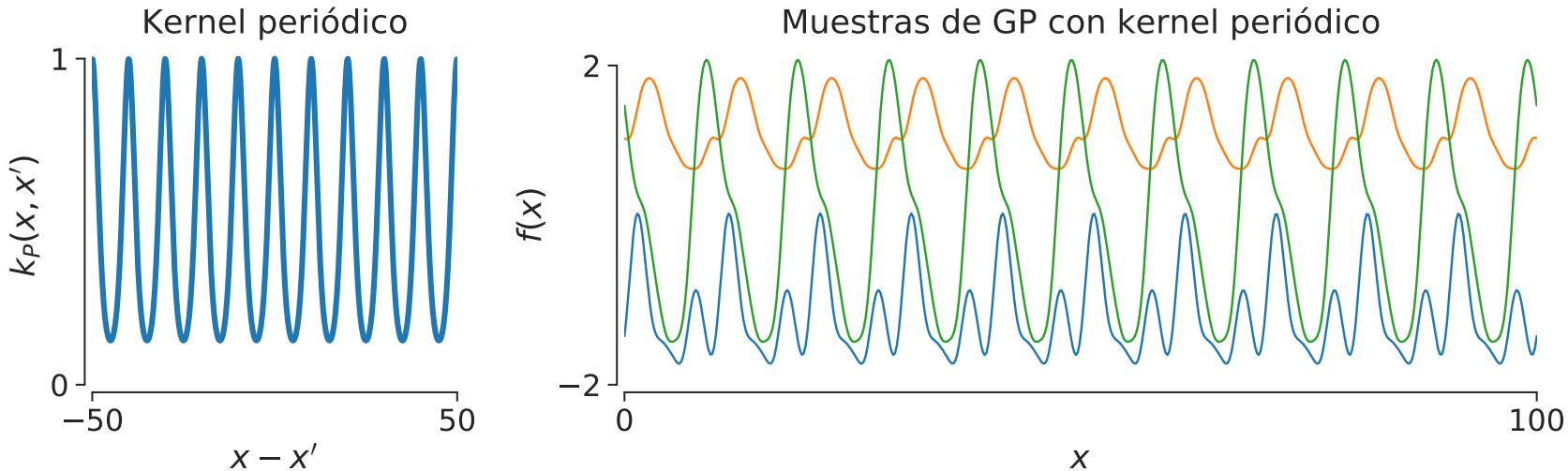

Fig.. Kernel periódico, en la izquierda se muestra la covarianza en función de su argumento $\tau = x - x'$, a la derecha de un $\mathcal{GP}$ usando un kernel periódico.

Extensiones de un $\mathcal{GP}$ - Clasificación

Hemos visto como usar un $\mathcal{GP}$ para regresión, pero este también puede usado para clasificación, para esto simplemente "pasamos" nuestro $\mathcal{GP}$ por una función logística, para así obtener un prior sobre $\sigma\left(f(x)\right)$ donde σ es la función logística. Sin embargo esto trae consigo un problema, pues ahora la distribución posterior a las observaciones no se tiene de forma analítica como para el caso de regresión, esto lleva a que tengamos que recurrir a métodos aproximado de inferencia. Una solución simple es utilizar la aproximación de Laplace, pero si se quieren aproximaciones más fidedignas métodos más complejos pueden ser usados como *Expectation Maximization* y métodos MCMC.

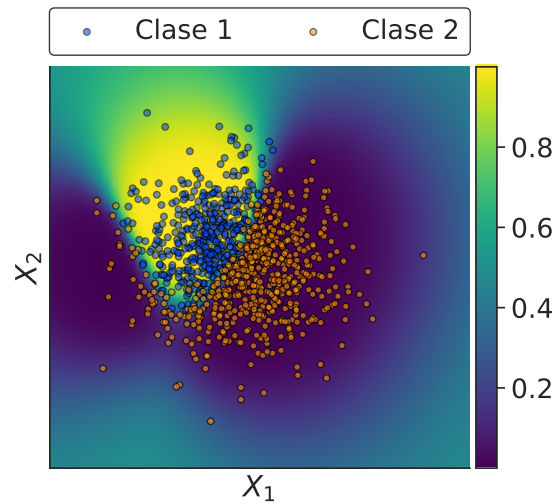

Fig.. $\mathcal{GP}$ de clasificación utilizando datos sintéticos. Este clasificador entrega una densidad de probabilidad en vez de una sola función de decisión.

Extensiones de un $\mathcal{GP}$ - Selección automática de features

Un $\mathcal{GP}$ define una densidad de probabilidad sobre funciones, donde estas funciones son del tipo $f : \mathbb{R}^D \to \mathbb{R}$, con D es finito, este es nuestra dimensión de entrada o "características". Haciendo un pequeño cambio en nuestra función kernel podemos hacer que esta automáticamente seleccione las entradas más relevantes con el problema, es decir realice una selección de características automática.

Consideremos el siguiente Kernel

$$k(x, x') = \sigma^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2\ell_d^2}\right)$$

vemos que es una multiplicación de kernels SE, donde se tiene un *lenghtscale* por cada entrada ℓ_d, sabemos que mientras más grande es este ℓ_d menos flexible será el $\mathcal{GP}$ respecto a cambios en ese eje, haciendo que las funciones del proceso dependan cada vez menos de la componente d a medida que $\ell_d \to \infty$. De esta forma se puede controlar de forma automática la relevancia de cada eje del conjunto de entrada, pues los parámetros del kernel se obtienen en el entrenamiento. De esta forma estamos optimizando también en que grado afecta cada variable en nuestra predicción.

Extensiones de un $\mathcal{GP}$ - Multi output $\mathcal{GP}$

Hasta el momento solo hemos hablado de $\mathcal{GP}$ cuando nuestro proceso es solo una dimensión de salida. Se pueden extender los procesos Gaussianos a funciones de más de una salida o canal, donde ahora la función de covarianza $k(x, x')$ no entrega un escalar sino una matriz definida positiva, donde la diagonal corresponde a la covarianza del canal o autocovarianza y los elementos fuera de la diagonal corresponden a las covarianzas cruzadas o cross-covarianza. Este tipo de procesos Gaussianos aumentan considerablemente de complejidad al diseñar funciones de covarianza.

Dado un número m de canales, se tendrán m funciones de autocovarianza y ahora $m(m-1)/2$ funciones de covarianza y $k(x, x')$ será una matriz de $m \times m$. El desafio está en diseñar o escoger estas funciones de tal forma que para cualquier par de puntos x, x' la matriz $k(x, x')$ sea definida positiva.

Una opción simple es asumir que los canales son independientes entre sí, lo que equivale a entrenar independientemente m procesos Gaussianos, uno para cada canal, esto facilita el diseño de las funciones de covarianza pero hace que se pierdan relaciones entre los canales.

Clase 21 - Procesos Gaussianos II

Aprendizaje de Máquinas - MA5204

Felipe Tobar

Department of Mathematical Engineering &
Center for Mathematical Modelling
Universidad de Chile

15 de junio de 2021

UNIVERSIDAD
DE CHILE