Regresión Lineal Funcional Bayesiana

Regresión Lineal Funcional Bayesiana

Marco teórico

Planteamiento Bayesiano

Modelo RKHS: $Y = \alpha_0 + \Psi_X^{-1}(\alpha) + \epsilon$, donde $\alpha \in \mathcal{H}_K$ y $\epsilon \sim \mathcal{N}(0, \sigma^2)$, i.e.:

$$Y_i \mid \theta, X_i = x_i \sim \mathcal{N}\left(\alpha_0 + \sum_{j=1}^{p} \beta_j x_i(\tau_j),\ \sigma^2\right).$$

Distribuciones a priori:

$$\begin{aligned}\pi(\alpha_0, \log\sigma) &\propto 1,\\ \tau &\sim \mathcal{U}([0,1]^p),\\ \beta \mid \tau, \sigma^2 &\sim \mathcal{N}\left(b_0, g\sigma^2\left[\mathcal{X}_\tau'\mathcal{X}_\tau + \eta\lambda_{\max}(\mathcal{X}_\tau'\mathcal{X}_\tau)\right]^{-1}\right),\end{aligned}$$

Log-posterior:

$$\begin{aligned}\log\pi(\beta, \tau, \alpha_0, \log\sigma \mid \boldsymbol{Y}) \propto{}& \frac{1}{2}\log|G_\tau| - (p+n)\log\sigma\\ &- \frac{1}{2\sigma^2}\left(\|\boldsymbol{Y} - \alpha_0\mathbf{1} - \mathcal{X}_\tau\beta\|^2 + \frac{1}{g}(\beta - b_0)' G_\tau (\beta - b_0)\right)\end{aligned}$$

Modelo Bayesiano

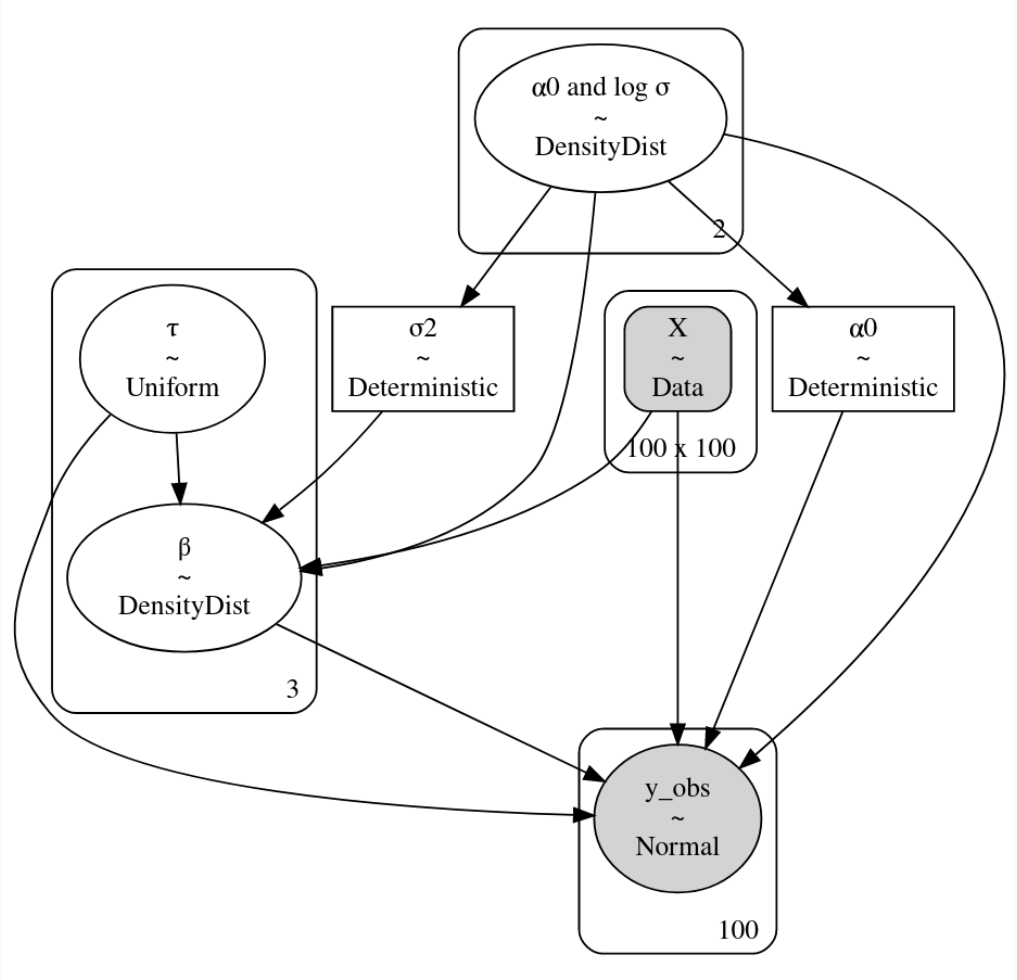

Figura 1: Relaciones entre los parámetros del modelo.

Aproximación de la posteriori

Procedimiento: Utilizar métodos MCMC para aproximar la distribución a posteriori. En concreto, se barajan tres alternativas:

- *Affine-Invariant Ensemble Sampler*. Conjunto de cadenas que se influencian mutuamente para dar cada paso.
- *NUTS*. Algoritmo que utiliza información del gradiente de la función objetivo para dar cada paso.
- *Metropolis*. Algoritmo estándar de *markov chain monte carlo*.

Los resultados son similares.

Algoritmo

Fijamos algunos hiperparámetros:

- Número de cadenas: 64.
- Número de pasos: $100 + 1000$.
- Movimientos: elección aleatoria (ponderada) entre dos de los recomendados, *stretch* y *walk*.
- Inicialización: Entorno aleatorio del MLE + muestras de las distribuciones a priori.
- Prior de β: b_0 se elige como el MLE de β, y fijamos $g = 5$ (recomendado en *Grollemund et al. (2019)*).

Model checking

- Análisis de la traza de las cadenas y de la distribución a posteriori de los parámetros. Se obtienen **intervalos creíbles** para los parámetros.
- En cada paso obtenemos una estimación $\tilde{\theta}_m$, y podemos generar $Y^{(m)*} \mid \theta_m, X$ siguiendo el modelo asumido.
- *Bayesian p-values*: $p = P(T(Y^*) \leq T(Y) \mid Y)$ para ciertas elecciones de T: mínimo, máximo, mediana, media. Se calcula contando la proporción de muestras generadas que cumplen la desigualdad, y se espera que esté en torno a 0.5.

Model checking II

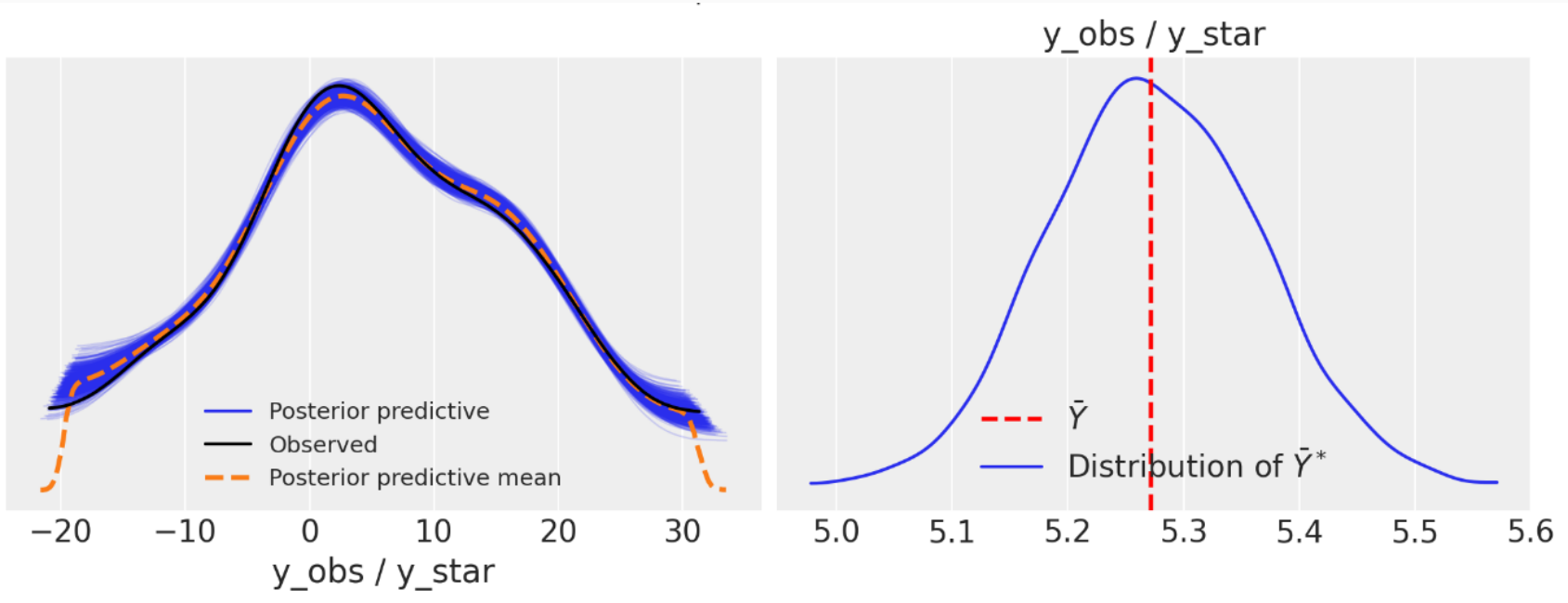

Figura 2: *Posterior predictive checks* bajo un modelo RKHS.

Predicciones

Estimación puntual: Se resume la distribución a posteriori de los parámetros $\theta \mid Y$ mediante un estimador puntual (media, mediana, moda), y se utilizan para predecir según el modelo:

$$\hat{Y}_i = \hat{\alpha}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_i(\hat{\tau}_j), \quad i = 1, \ldots, n.$$

Estimación distribucional: Se utiliza la media de *todas* las muestras generadas de Y^* como predicción:

$$\hat{Y} = \frac{1}{M} \sum_{m=1}^{M} Y^{(j)*}.$$

En ambos casos se puede considerar solo una de cada k muestras.

Predicciones II

Podemos hacer también una selección de p variables usando los estimadores puntuales de $\tau \mid Y$, y después aplicar cualquier algoritmo de regresión lineal.

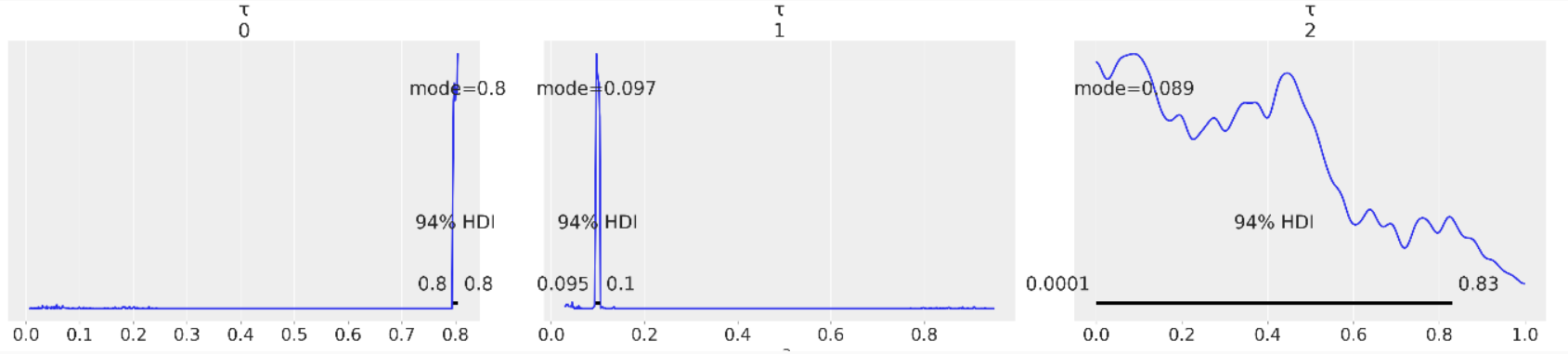

Figura 3: Distribución a posteriori estimada para los puntos de impacto.

En cualquier caso, medimos el error usando el MSE como métrica.

Regresión Lineal Funcional Bayesiana

Experimentos

Datos sintéticos

Se consideran 150 ejemplos de $X \sim GP(0, K(s,t))$ y tres posibles variantes de K: movimiento browniano fraccional ($H = 0.8$), Ornstein-Uhlenbeck y kernel RBF.

Se genera la respuesta Y acorde a dos modelos, RKHS y L^2. Concretamente, se elige

$$Y_i \sim \mathcal{N}\big(5 - 5X_i(0.1) + 10X_i(0.8),\ 0.5\big)$$

ó

$$Y_i \sim \mathcal{N}\left(5 + \int_0^1 \log(1+4t)X_i(t)\,dt,\ 0.5\right).$$

Consideramos una malla regular de $N = 100$ puntos en $[0,1]$, y un reparto de 100 ejemplos para entrenamiento y 50 para evaluación.

Datos reales

Se consideran dos conjuntos de datos reales.

- **Tecator:** Contiene 215 ejemplos de mediciones de *absorbancia* en muestras de carne para intentar predecir su contenido en grasa.
- **Aemet**: Contiene 73 ejemplos de curvas de temperatura, a partir de las cuales se intenta predecir la precipitación total.

En ambos casos hacemos una división $80\,\% - 20\,\%$ para entrenamiento y *test*.

Preprocesado de los datos

- Se centran los regresores para que tengan media 0. Opcionalmente, se pueden estandarizar en cada punto de la malla.
- Se permite estandarizar también la respuesta Y.
- Se permite **sustituir los datos X_i por su expansión en una base** de Fourier, con un número determinado de coeficientes. De esta forma realizamos un suavizado de las curvas.

Algoritmos de comparación

Regresión lineal multivariante: Regresión Lineal estándar, Lasso (L^1), Ridge (L^2).

Regresión no lineal multivariante: SVM con kernel RBF.

Regresión lineal funcional: Modelo L^2, KNN Funcional.

Reducción de dimensión: PCA Funcional (proyección sobre coeficientes).

Selección de variables: Aleatoria.

Se entrenan todos ellos sobre el conjunto de entrenamiento, haciendo *5-fold cross validation* para escoger los mejores hiperparámetros en cada caso (valor de regularización, número de componentes, ...).

Metodología experimental

- Para cada conjunto de datos consideramos tres posibles valores de p y cuatro posibles valores de η:
  - $p \in \{2, 3, 4\}$ para los datos sintéticos y $p \in \{3, 4, 5\}$ para los datos reales.
  - $\eta \in \{0.01, 0.1, 1.0, 10.0\}$.
- Para aumentar la estabilidad hacemos 5 repeticiones de cada modelo. Es decir, entrenamos un total de 60 modelos sobre el conjunto de entrenamiento
- Con el mejor modelo obtenido se realiza también una selección de variables basada en los valores de τ estimados mediante los distintos estimadores puntuales.

Además, repetimos este proceso con los datos suavizados en una base de Fourier con 11 elementos (también con los algoritmos de comparación).

Resultados RKHS

Sin base

Kernel	Victoria	Victoria sel. variables
fBM	✓	✓
O-U	✓	✓
RBF	✓	✓

Base Fourier(11)

Kernel	Victoria	Victoria sel. variables
fBM	✓	✓
O-U	✓	✗
RBF	✓	✓

Resultados L^2

Sin base

Kernel	Victoria	Victoria sel. variables
fBM	✗	✗
O-U	✗	✗
RBF	✗	✗

Base Fourier(11)

Kernel	Victoria	Victoria sel. variables
fBM	✓	✓
O-U	✓	✓
RBF	✗	✗

Resultados datos reales

Sin base

Dataset	Victoria	Victoria sel. variables
Tecator	✗	✗
Aemet	✗	✓

Base Fourier(11)

Dataset	Victoria	Victoria sel. variables
Tecator	✗	✗
Aemet	✗	✓

Algunas observaciones

- Por cada modelo entrenado obtenemos 4 estimadores y 3 estrategias de selección de variables.
- En general, se obtienen mejores resultados con el uso de la base.
- El tiempo medio de entrenamiento de cada modelo es de unos 20-30 segundos.
- En ocasiones, nuestro algoritmo supera con holgura a todos los algoritmos de comparación. Cuando pierde, suele ser frente a uno o dos de ellos (y por poco), pero sigue superando al resto.
- Se puede considerar como algoritmo de comparación el estimador de máxima verosimilitud de los parámetros. En casi todas las pruebas experimentales, el desempeño de nuestro algoritmo supera al del MLE.

Dificultades

- Hay multitud de grados de libertad: estandarización de regresores y/o variable respuesta, elección o no de una base (¿cuál?,) algoritmo de estimación del MLE, número, longitud y movimientos de las cadenas MCMC, elección de distribuciones a priori, elección de g, b_0 y η, etc.
- El algoritmo MCMC es costoso, y en conjuntos de datos grandes las estrategias de *cross-validation* para seleccionar parámetros pueden no ser viables.
- Es necesario un procedimiento para seleccionar el valor de p (BIC, WAIC, WBIC, *cross-validation*, ...).
- Debido a la aleatoriedad del algoritmo, los resultados pueden variar sustancialmente de una ejecución a otra.
- La distribución a priori para β depende en gran medida del valor de escogido. Si su estimación inicial no es buena, el algoritmo no funciona demasiado bien.

Dificultades II

- Hay un problema de *identificabilidad* de los coeficientes (son intercambiables), que se agudiza especialmente al usar varias cadenas.
- En el caso en el que el modelo subyacente es RKHS, es posible que no se recuperen los verdaderos valores de los parámetros debido a interacciones entre los coeficientes (si p es mayor que el número real de componentes).

Alternativas

- Explorar el concepto de *online learning* aplicado a esta situación, por ejemplo usando como priori de nuevos datos la posteriori aprendida.
- Utilizar otras herramientas de suavizado en lugar de bases de Fourier.
- Sustitutir la distribución a priori de β por otra que requiera menos hiperparámetros.

Regresión Logística Funcional Bayesiana

Regresión Logística Funcional Bayesiana

Marco teórico

Planteamiento Bayesiano

Modelo: Cada Y_i se puede ver como una variable aleatoria de Bernoulli $\mathcal{B}(p(x_i))$, con

$$p_i \equiv p(x_i) = \mathbb{P}(Y_i = 1 \mid X_i = x_i) = \frac{1}{1 + \exp\left\{-\alpha_0 - \sum_{j=1}^{p} \beta_j x_i(\tau_j)\right\}}.$$

Distribuciones a priori: Igual que en regresión lineal.

Log-posterior:

$$\log \pi(\beta, \tau, \alpha_0, \log\sigma \mid Y) \propto$$
$$\sum_{i=1}^{n} \left[\left(\alpha_0 + \Psi_{x_i}^{-1}(\alpha)\right) y_i - \log\left(1 + \exp\left\{\alpha_0 + \Psi_{x_i}^{-1}(\alpha)\right\}\right)\right]$$
$$+ \frac{1}{2}\log|G_\tau| - p\log\sigma - \frac{1}{2g\sigma^2}(\beta - b_0)' G_\tau (\beta - b_0).$$

Model checking

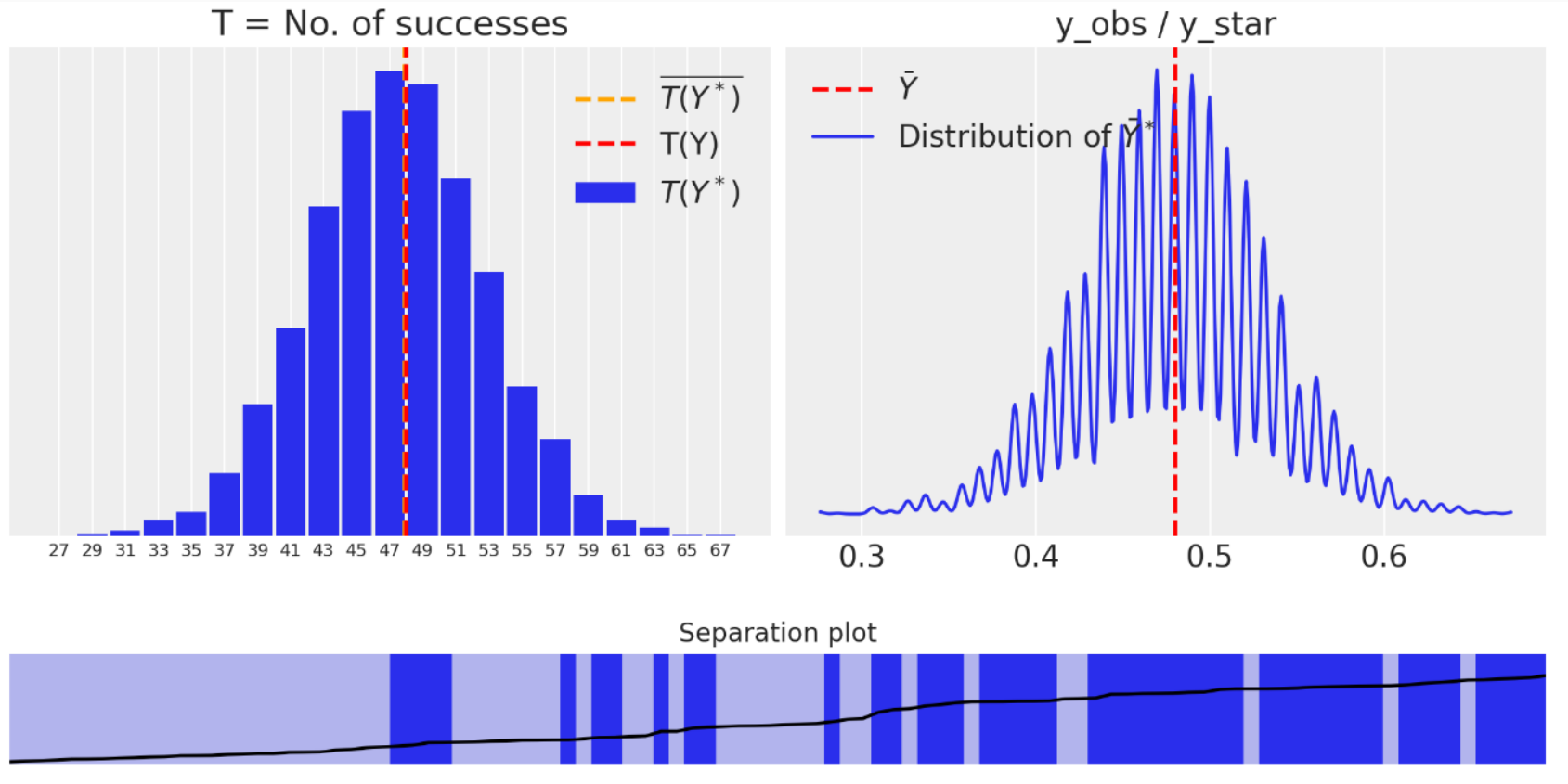

Figura 4: *Posterior predictive checks* bajo un modelo RKHS.

Predicciones

- Establecemos un umbral en 0.5 para establecer la pertenencia a las clases.
- Los estimadores puntuales y la selección de variables son análogos al caso de regresión lineal.
- Tenemos ahora dos estimadores basados en la distribución a posteriori:
  - Basado en el voto mayoritario de las muestras Y^* generadas.
  - Basado en la media de las probabilidades p_i^* generadas (con aplicación posterior del umbral).
- Se usa el *accuracy* (precisión) para evaluar los modelos.

Regresión Logística Funcional Bayesiana

Experimentos

Datos sintéticos

- Misma estrategia de generación que antes. Se elige

$$Y_i \sim \mathcal{B}\left(\frac{1}{1+\exp\left\{0.5+5X_i(0.1)-10X_i(0.8)\right\}}\right)$$

ó

$$Y_i \sim \mathcal{B}\left(\frac{1}{1+\exp\left\{0.5-\int_0^1 \log(1+4t)X_i(t)\,dt\right\}}\right).$$

- Añadimos una nueva estrategia: generar datos procedentes de dos procesos distintos, y etiquetarlos acorde a su verdadera distribución.
  - Browniano vs. browniano fraccional.
  - RBF vs. Ornstein-Uhlenbeck.
  - RBF(0.2) vs. RBF(0.5).

Datos reales

Se consideran dos conjuntos de datos reales.

- **Medflies:** Contiene 534 ejemplos de mediciones del número de huevos diario puesto por una serie de moscas, para intentar predecir si viven mucho o poco.
- **Growth**: Contiene 93 ejemplos de curvas de altura en niños y niñas.

En ambos casos hacemos una división $80\,\% - 20\,\%$ para entrenamiento y *test*.

Algoritmos de comparación

Clasificación lineal multivariante: Regresión Logística, SVM lineal.

Clasificación no lineal multivariante: SVM con kernel RBF.

Clasificación funcional: *Maximum Depth*, *Nearest Centroid* Funcional, KNN Funcional.

Reducción de dimensión: PCA Funcional (proyección sobre coeficientes).

Selección de variables: Aleatoria, *Recursive Maxima Hunting*, *RKVS* (Mahalanobis).

Mismo preprocesado y metodología experimental que en el caso de regresión lineal, utilizando esta vez 9 coeficientes de Fourier.

Resultados RKHS

Sin base

Kernel	Victoria	Victoria sel. variables
fBM	✓	✗
O-U	✗	✓
RBF	✓	✗

Base Fourier(9)

Kernel	Victoria	Victoria sel. variables
fBM	✓	✓
O-U	✓	✓
RBF	✓	✓

Resultados L^2

Sin base

Kernel	Victoria	Victoria sel. variables
fBM	✗	✗
O-U	✓	✗
RBF	✓	✓

Base Fourier(9)

Kernel	Victoria	Victoria sel. variables
fBM	✗	✓
O-U	✗	✓
RBF	✓	✓

Resultados MIXTURE

Sin base

Kernel	Victoria	Victoria sel. variables
fBM + BM	✗	✗
O-U + RBF	✗	✗
RBF(0.2) + RBF(0.5)	✗	✗

Base Fourier(9)

Kernel	Victoria	Victoria sel. variables
fBM + BM	✗	✗
O-U + RBF	✗	✗
RBF(0.2) + RBF(0.5)	✗	✓

En este caso, los resultados de nuestro algoritmo son bastante malos (peor que un algoritmo aleatorio).

Resultados datos reales

Sin base

Dataset	Victoria	Victoria sel. variables
Medflies	✗	✗
Growth	✓	✓

Base Fourier(9)

Dataset	Victoria	Victoria sel. variables
Medflies	✗	✗
Growth	✓	✓

Tanto en las victorias como en las derrotas, la precisión es muy similar. Podemos considerar un empate en ambos conjuntos.