Resultados actualizados y otros algoritmos de comparación

Regresión y clasificación funcional

Antonio Coín 25 de enero de 2022

Universidad Autónoma de Madrid Departamento de Matemáticas



Partial Least Squares (PLS)

PLS multivariante

Se trata de un método supervisado. Descomponemos

$$X = TP^T, \quad Y = UQ^T$$

de forma que se maximice la covarianza entre X e Y (algoritmo PLS1).

Una vez aprendida la transformación, podemos realizar regresión lineal de dos maneras:

- Utilizando las transformaciones de X e Y para expresar $Y = X\beta$ a partir de ellas (PLS Regression).
- Transformando los datos X_i ← T_i y aplicando cualquier método de regresión multivariante sobre el conjunto ({T_i, Y_i}), donde ahora T_i son datos de dimensión posiblemente menor (reducción de dimensionalidad).

1

PLS Funcional

El algoritmo PLS multivariante se puede aplicar directamente a los datos funcionales (discretizados) para reducir la dimensión de los mismos. Sin embargo, existe una extensión puramente funcional (Preda & Saporta, 2001) que se basa en el siguiente problema de optimización:

$$\max_{w,c} \operatorname{Cov}^2 \left(\int_{\mathcal{T}} X(t) w(t) \, dt, cY \right).$$

Se define la primera componente principal PLS como la variable aleatoria

$$t_1 = \int_{\mathcal{T}} X(t)w(t) dt,$$

y se sigue un proceso iterativo análogo al caso finito-dimensional para encontrar el resto de componentes.

Implementación PLS funcional I

Caso I (extensión multivariante): utilizar un desarrollo en base.

Supongamos que tenemos los datos desarrollados en una base $X_i(t) = \sum_j \alpha_{ij} \phi_j(t)$. En Aguilera et al. (2010) se demuestra que la regresión PLS funcional de Y sobre X es equivalente a la regresión PLS multivariante de Y sobre $A\Phi^{1/2}$, donde $A=(\alpha_{ij})$ es la matriz de coeficientes y Φ es la matriz de Gram de los elementos de la base.

Implementación: casi inmediata usando librerías de regresión PLS multivariante.

Implementación PLS funcional II

Caso II (puramente funcional): obtener una base de componentes PLS directamente a partir de los datos.

Se trata del algoritmo APLS (Delaigle & Hall, 2012). Presenta un enfoque alternativo donde se construyen secuencialmente h funciones ortonormales $\{\phi_p(t)\}_{p=1,\dots,h}$ para maximizar

$$\operatorname{\mathsf{Cov}}\left(\mathsf{Y}-\mathsf{g}_{p-1}(\mathsf{X}),\int_{\mathcal{T}}\mathsf{X}\phi_{p}\right),$$

donde $g_p(x) = \sum_{j=1}^p \beta_j \int_{\mathcal{T}} x \phi_j$. Los valores óptimos de β se encuentran empíricamente por mínimos cuadrados. Finalmente, predecimos con g_h .

Implementación: es necesario aproximar integrales numéricamente y utilizar el algoritmo de Gram-Schmidt.

4

LDA y regresión lineal

LDA

El algoritmo de *Linear Discriminant Analysis* para el caso de clasificación binaria es equivalente¹ a la regresión lineal por mínimos cuadrados (usando un umbral de decisión). Lo mismo ocurre con sus respectivas extensiones al caso funcional, que en el caso de LDA persigue encontrar combinaciones lineales (en L^2) $\Phi(X)$ de los predictores para maximizar

$$\frac{\mathsf{Var}(\mathbb{E}[\Phi(X)\mid Y])}{\mathsf{Var}(\Phi(X))},$$

y en el caso de la regresión lineal trata de modelar la relación $Y = \Phi(X) + \varepsilon$. La regla de clasificación resultante para un umbral t es $g(x) = \mathbb{I}(\Phi(x) > t)$. Este umbral de decisión se fija de facto a la media ponderada de las respuestas codificadas $Y = \{y_0, y_1\}$:

$$t = p_0 y_0 + p_1 y_1$$
, con $p_i = n_i / n$.

¹Salvo por el término independiente; ver Gareth et al. (2013), Hart et al. (2000) y Mika (2003).

Regresión PLS para clasificación

En el caso funcional, el uso de regresión lineal para clasificación presenta algunos problemas (Preda & Saporta, 2005, y referencias contenidas). En este mismo artículo se propone usar la regresión PLS para aproximar la función discriminante.

Algoritmo:

- 1. Recodificar las etiquetas $0 \longleftrightarrow -\sqrt{p_1/p_0}$ y $1 \longleftrightarrow \sqrt{p_0/p_1}$.
- 2. Realizar regresión PLS de Y sobre X para obtener $\Phi_{PLS}(X) = \int_{\mathcal{T}} X(t) \beta_{PLS}(t) dt$.
- 3. Clasificar en la clase 0 si $\mathbb{I}(\Phi_{PLS}(x) < 0)$.

Resultados

Nota sobre las implementaciones

En una búsqueda rápida no se encontraron ninguno de los métodos funcionales comentados anteriormente implementados directamente en Python. Se ha hecho una implementación vectorizada (intentando maximizar la eficiencia) y al estilo del paquete *scikit-fda*, lo que permitiría fácilmente su integración en el mismo.

Resultados

Regresión lineal

Resultados RKHS kernel fBM

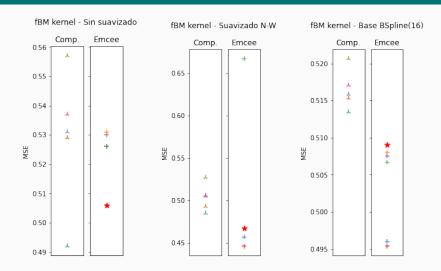


Figura 1: MSE de los mejores estimadores en cada caso bajo el modelo subyacente RKHS con kernel Browniano fraccional.

Resultados RKHS kernel O-U

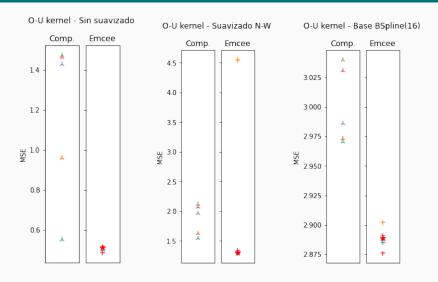


Figura 2: MSE de los mejores estimadores en cada caso bajo el modelo subyacente RKHS con kernel de Ornstein-Uhlenbeck.

Resultados RKHS kernel RBF

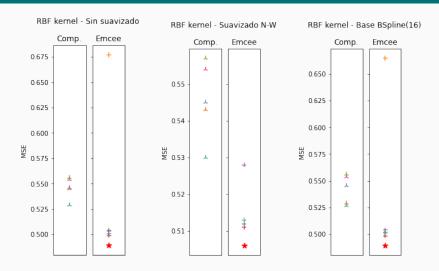


Figura 3: MSE de los mejores estimadores en cada caso bajo el modelo subyacente RKHS con kernel RBF.

Resultados L² kernel fBM

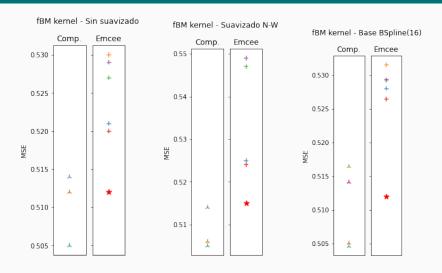


Figura 4: MSE de los mejores estimadores en cada caso bajo el modelo subyacente L^2 con kernel Browniano fraccional.

Resultados L² kernel O-U

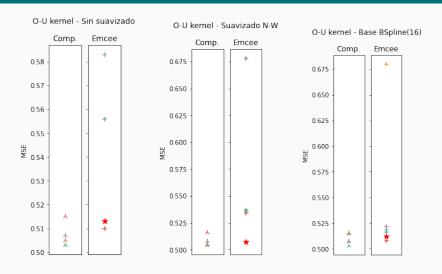


Figura 5: MSE de los mejores estimadores en cada caso bajo el modelo subyacente L^2 con kernel de Ornstein-Uhlenbeck.

Resultados L² kernel RBF

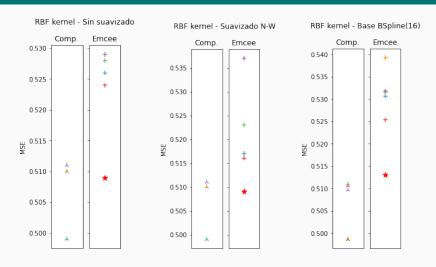


Figura 6: MSE de los mejores estimadores en cada caso bajo el modelo subyacente L^2 con kernel RBF.

Resultados Tecator

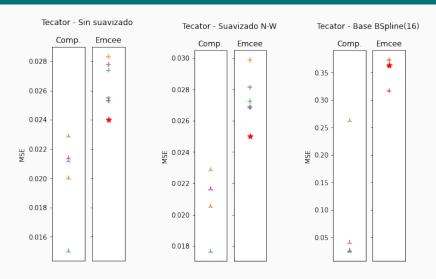


Figura 7: MSE de los mejores estimadores en cada caso para el conjunto de datos Tecator.

Resultados Aemet

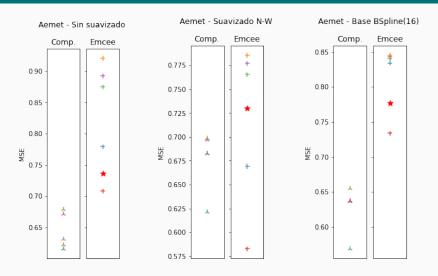


Figura 8: MSE de los mejores estimadores en cada caso para el conjunto de datos Aemet.

Resultados

Regresión logística

Resultados RKHS kernel fBM

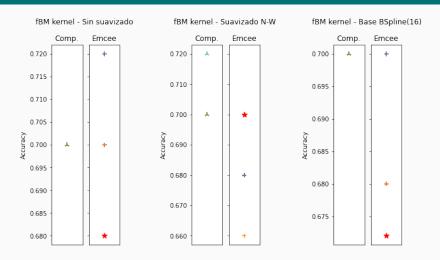


Figura 9: Accuracy de los mejores estimadores en cada caso bajo el modelo subyacente RKHS con kernel Browniano fraccional.

Resultados RKHS kernel O-U

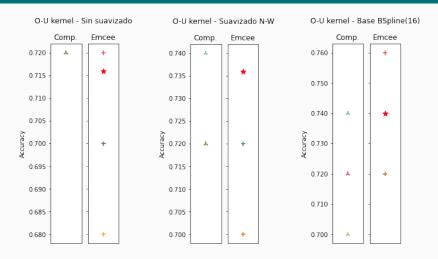


Figura 10: Accuracy de los mejores estimadores en cada caso bajo el modelo subyacente RKHS con kernel de Ornstein-Uhlenbeck.

Resultados RKHS kernel RBF

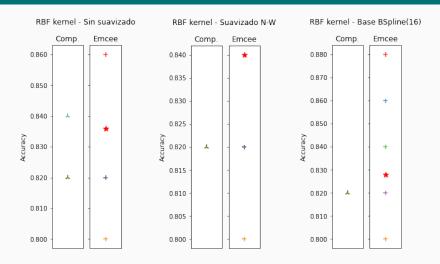


Figura 11: Accuracy de los mejores estimadores en cada caso bajo el modelo subyacente RKHS con kernel RBF.

Resultados L² kernel fBM

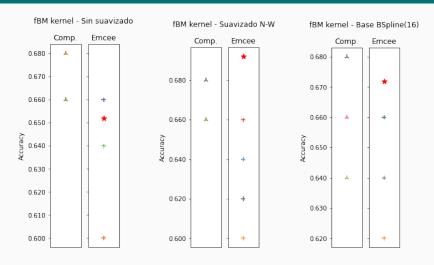


Figura 12: Accuracy de los mejores estimadores en cada caso bajo el modelo subyacente L^2 con kernel Browniano fraccional.

Resultados L² kernel O-U

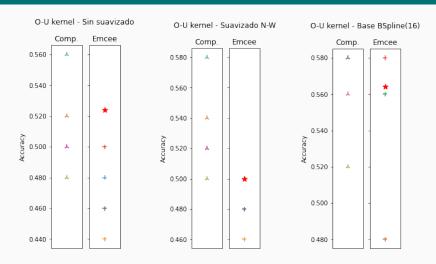


Figura 13: Accuracy de los mejores estimadores en cada caso bajo el modelo subyacente L^2 con kernel de Ornstein-Uhlenbeck.

Resultados L² kernel RBF

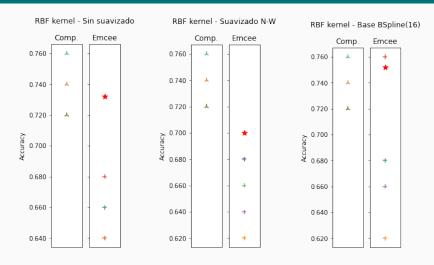


Figura 14: Accuracy de los mejores estimadores en cada caso bajo el modelo subyacente L^2 con kernel RBF.

Resultados Medflies

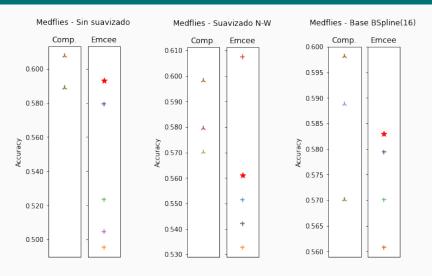


Figura 15: Accuracy de los mejores estimadores en cada caso para el conjunto de datos Medflies.

Resultados Growth

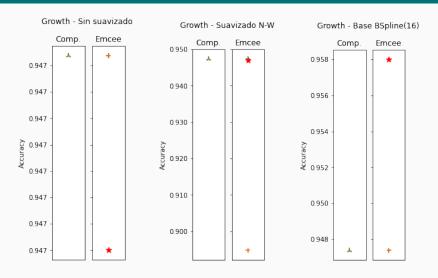


Figura 16: Accuracy de los mejores estimadores en cada caso para el conjunto de datos Growth.

Referencias

Bibliografía

- Aguilera, A. M., Escabias, M., Preda, C., & Saporta, G. (2010). Using basis expansions for estimating functional PLS regression: applications with chemometric data. *Chemometrics and Intelligent Laboratory Systems*, 104(2), 289-305.
- Delaigle, A., & Hall, P. (2012). Methodology and theory for partial least squares applied to functional data. *The Annals of Statistics*, 40(1), 322-352.
- Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2013). *An introduction to statistical learning: with applications in R.* Spinger.
 - Hart, P. E., Stork, D. G., & Duda, R. O. (2000). *Pattern classification*. Wiley Hoboken.
- Mika, S. (2003). Kernel fisher discriminants.
 - Preda, C., & Saporta, G. (2001). PLS regression on a stochastic process. PLS'01. Second International Symposium on PLS and related methods. https://hal.archives-ouvertes.fr/hal-01124683
 - Preda, C., & Saporta, G. (2005). PLS discriminant analysis for functional data. ASMDA'05 XIth Int. Symp. on Applied Stochastic Models