Aplicaciones del *stable rank* en el marco del análisis topológico de datos

Daniel Miguel
damigutr@unirioja.es
Dpto. de Matemáticas y Computación, Universidad de La Rioja

Palabras clave: homología persistente, *stable rank*, análisis topológico de datos

Introducción

Dado un conjunto de datos, podemos dotarlo de una topología definiendo sobre él una pseudométrica. En el caso de la homología persistente, definimos además una filtración sobre el complejo simplicial total en los puntos del dataset. Una vez hecho esto, podemos estudiar cómo evolucionan las características homológicas de dicho complejo en los diferentes niveles de la filtración, e intentar averiguar si tienen alguna relevancia en el contexto del que provengan los datos. Los objetos resultantes de este análisis se conocen como **módulos de persistencia** y, en el caso 1–dimensional, son usualmente representados mediante **diagramas de persistencia** o ***barcodes***.

Una vez construidos los *barcodes*, tenemos diferentes distancias (como Wasserstein o Bottleneck) que nos permiten compararlos entre sí y, por ejemplo, clasificar datos o aplicar procesos de *clustering*. La utilidad de estos invariantes está ya sobradamente probada, pero también se plantean varios problemas: en el espacio de diagramas de persistencia es difícil definir y utilizar herramientas estadísticas, los diagramas no están definidos para el caso de la multipersistencia, y las distancias Bottleneck y Wasserstein pueden llegar a pasar por alto aspectos significativos de éstos.

El ***stable rank*** es un invariante topológico introducido en [1] que, partiendo de un módulo de (multi-)persistencia, construye una función no decreciente y constante por partes. Esta transformación conlleva una pérdida de información, pero ésta pérdida se puede controlar mediante la elección de

Parcialmente financiado por la Ayuda PID2020-116641GB-I00 financiada por MCIN/AEI/10.13039/501100011033, así como por el proyecto Inicia 2023/02 financiado por el Gobierno de La Rioja (España)

ciertos parámetros. Se puede comparar con otros invariantes como los ***persistence images*** o ***persistent landscapes***, en tanto que trata de resolver el mismo problema. En la charla veremos algunos ejemplos de aplicaciones del *stable rank*. Para ver un pipeline general con los aspectos fundamentales de las aplicaciones, se puede consultar [2].

Aplicaciones

- *Point processes*, [3]: Se generan 500 simulaciones de 6 clases diferentes de *point processes* en el cuadrado unidad. 200 de ellas se utilizan para generar los *stable ranks* representativos de cada una de las clases (haciendo la media). El problema consiste en clasificar las 300 restantes comparando sus propios *stable ranks* con los representantes de cada clase.

- *Activity monitoring*, [4]: Se toma el conjunto de datos PAMAP2 ([5]), que consiste en series de datos temporales indexados según actividad física e individuo. Para cada par de ellos, se toman muestras sobre los datos uniformemente y sin reemplazo. Estos datos sirven de nuevo como clasificadores, pero en este caso se utilizan para construir *stable rank kernels* e integrarlos en modelos de *machine learning* (SVM). Finalmente, el modelo se entrena y se testea, evaluando su precisión.

- Aprendizaje semi–supervisado: Este último ejemplo es parte de un trabajo en marcha del ponente, que fue parcialmente presentado en [6] y que retoma un problema presentado originalmente en [7]. Nuestro *input* es un conjunto de datos dentro del cual tenemos algunos puntos ya etiquetados en dos clases (ceros y unos). El objetivo es extender el etiquetado al resto del conjunto.

  Para cada punto ya etiquetado p, construímos dos *stable ranks*: tomamos 100 muestras sobre el conjunto de ceros (unos), muestreando en función de una distribución normal centrada en p, y calculamos el *stable rank* del espacio obtenido. Después hacemos la media. Ahora tomamos un punto sin etiquetar q, y hacemos lo mismo. Comparamos estas funciones con las de los puntos ya etiquetados, y asignamos la etiqueta en función de los puntos etiquetados que tengan los *stable ranks* más parecidos.

  Probamos los modelos para diferentes datasets y clasificadores (*support vector machine* y *random forest*) y estudiamos la influencia de los diferentes parámetros que se pueden elegir a la hora de calcular los diferentes stable ranks.

Bibliografía

[1] Martina Scolamiero, Wojciech Chachólski, Anders Lundman, Ryan Ramanujam, and Sebastian Öberg. Multidimensional persistence and noise. *Foundations of Computational Mathematics*, 17:1367–1406, 2017.

[2] Jens Agerberg, Wojciech Chacholski, and Ryan Ramanujam. Data, geometry and homology. *arXiv preprint arXiv:2203.08306*, 2022.

[3] Henri Riihimäki and Wojciech Chacholski. Generalized persistence analysis based on stable rank invariant. *arXiv preprint arXiv:1807.01217*, 2018.

[4] Jens Agerberg, Ryan Ramanujam, Martina Scolamiero, and Wojciech Chachólski. Supervised learning using homology stable rank kernels. *Frontiers in Applied Mathematics and Statistics*, 7:668046, 2021.

[5] Attila Reiss. PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5NW2H.

[6] Ana Romero Daniel Miguel, Andrea Guidolin. Stable rank: an application to semi-supervised learning. `https://sites.google.com/view/tda2024/posters`, 2024. [Online; consultado el 21-Marzo-2024].

[7] Adrián Inés, César Domínguez, Jónathan Heras, Gadea Mata, and Julio Rubio. Semi-supervised machine learning: a homological approach. *Proceedings of the XVII EACA*, pages 109–112, 2022.