

Entropía de Permutación para Alerta Sísmica (baseline scikit-learn)

Martín Ramírez Espinosa, Sergio Alejandro González Osorio

December 10, 2025

¿Por qué entropía de permutación?

- Las formas de onda sísmicas cambian su complejidad antes de eventos energéticos; PE captura esto con pocas suposiciones.
- Ligero: sólo cuenta patrones ordinales, sin filtrado pesado ni features complejas.
- Robusto: desempate determinista mantiene PE estable en datos cuantizados o con bajo SNR.

Teoría: PE, WPE, MPE

Para dimensión de embedding m y retardo τ :

- ① Formar patrones ordinales desde $[x_t, x_{t+\tau}, \dots, x_{t+(m-1)\tau}]$ ordenando por valor y luego índice.
- ② Estimar la distribución de patrones p y calcular $H = -\sum p \log p$.
- ③ Normalizar por $\log(m!)$ para mantener $PE \in [0, 1]$.

WPE multiplica los conteos por la varianza local (penaliza ventanas planas). MPE repite sobre varios τ para capturar dinámica multiescala.

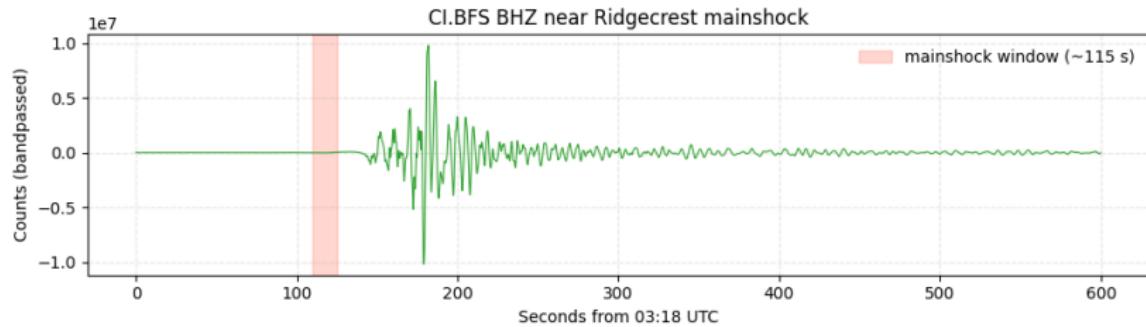
Implementación en este repo

- Algoritmos núcleo: `src/permuation_entropy/features.py` (PE/WPE/MPE, patrones deterministas, ventanas deslizantes).
- CLI de extracción: `data_features.ingest` (MiniSEED → CSV de features PE con tiempos de inicio alineados).
- Helper de modelo: `permuation_entropy.models` (regresión logística balanceada, métricas, probabilidades).
- CLI de entrenamiento: `bin/train_pe.py` (lee CSV, split estratificado cuando se puede, guarda gráfico de probabilidades).

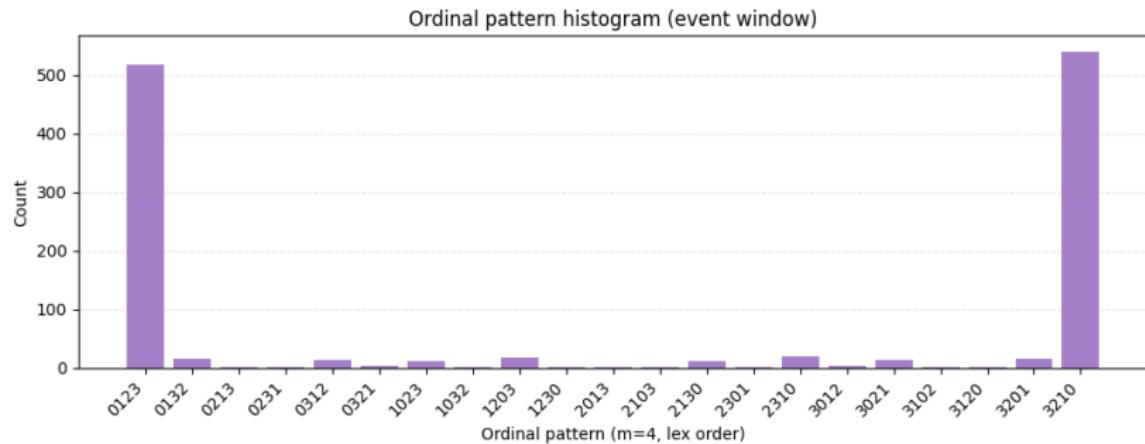
Caso de estudio: sismo principal de Ridgecrest

- Datos: MiniSEED de IRIS alrededor del sismo 2019-07-06 (IU.ANMO, CI.BFS para vista cercana).
- Ventanas: 30 s de largo, salto de 5 s; etiquetas marcan 1180–1220 s desde 03:00 UTC.
- Features: PE, WPE, MPE ($\tau = 1.4$); se reetiquetan si faltan en el CSV.
- Modelo: regresión logística balanceada; evaluación en split hold-out para evitar métricas optimistas.

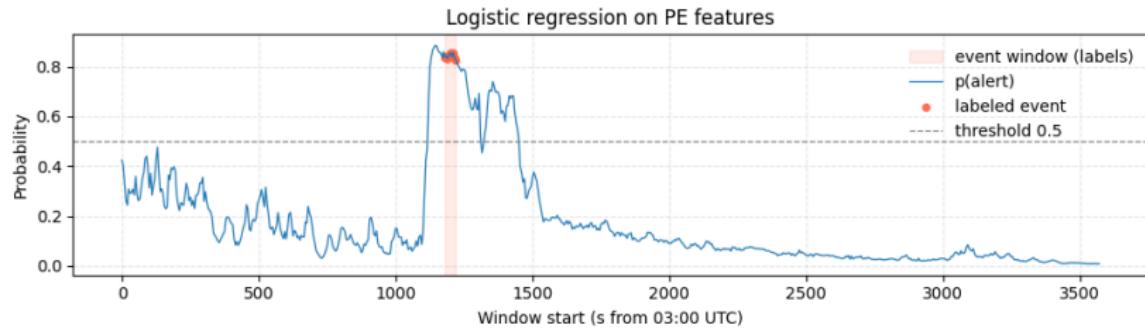
Vista de forma de onda (CI.BFS, estación cercana)



Patrones ordinales en una ventana



Probabilidades en el tiempo (ejemplo)



Resumen del pipeline

- ① Descargar o cargar MiniSEED; resolver rutas de datos automáticamente en notebooks/CLI.
- ② Segmentar la señal en ventanas solapadas; calcular PE/WPE/MPE por ventana.
- ③ Guardar CSV con start_sec para gráficos y label para entrenamiento supervisado.
- ④ Entrenar regresión logística; graficar probabilidades vs. tiempo para inspeccionar separabilidad.

Conclusiones y siguientes pasos

- Las features basadas en PE son un baseline rápido y transparente para alertas de erupción/temblor.
- Patrones deterministas y regresión balanceada reducen fragilidad en datasets pequeños.
- Siguiente: añadir validación cruzada, calibración de umbrales y features espectrales; empaquetar para pip + CI.