Clase 193 — Stack bayesiano moderno: PyMC v5, NumPyro, ArviZ

Parte: 3 — Estadística Inferencial y Causal · Fuente: PyMC v5 docs + NumPyro docs + ArviZ docs. Duración estimada: 90 min.

Clase 193 — Stack bayesiano moderno: PyMC v5, NumPyro, ArviZ

> *Parte: 3 — Estadística Inferencial y Causal · Fuente: PyMC v5 docs + NumPyro docs + ArviZ docs. Duración estimada: 90 min.*

Objetivo

Aprender el stack bayesiano moderno post-Theano —PyMC v5 (PyTensor), NumPyro (JAX), ArviZ (visualización backend-agnóstica)— a nivel de poder construir modelos jerárquicos serios, diagnosticar convergencia (r_hat, ess_bulk, divergences), comparar modelos con LOO-CV y WAIC, y elegir backend según escala.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Construir modelo jerárquico no-centered en PyMC v5.
- Diagnosticar: r_hat ≤ 1.01, ess_bulk ≥ 400, divergences = 0.
- Aplicar non-centered parameterization para evitar funnel posteriors.
- Comparar modelos con az.compare([m1, m2], ic='loo').
- Migrar modelo de PyMC a NumPyro para 10-50× speedup en CPU.
- Aplicar SVI (Stochastic Variational Inference) en NumPyro como alternativa rápida a MCMC.

Temas

- PyMC v5: PyTensor backend, sintaxis estable.
- NumPyro: JAX backend, JIT + autograd + GPU/TPU.
- Centered vs non-centered parametrization.
- Posterior predictive check con ArviZ.
- LOO-CV y WAIC para comparación.
- SVI con AutoNormal / AutoMultivariateNormal.

Definiciones y características

- PyTensor: sucesor de Theano (archivado 2017). Backend nativo de PyMC.
- NumPyro: probabilistic programming en JAX. 5-50× más rápido en CPU, GPU/TPU nativo.
- r_hat: Gelman-Rubin convergence. ≤ 1.01 → OK.
- ess_bulk / ess_tail: effective sample size. ≥ 400 bulk para inferencia.
- Non-centered: $x = \mu + \sigma \cdot z$, $z \sim N(0,1)$ en vez de $x \sim N(\mu, \sigma)$. Evita funnel.
- LOO-CV (Vehtari): leave-one-out con Pareto smoothed importance sampling.
- SVI: aproximación variational; cierra el gap MCMC con velocidad.

Dataset / recursos

- tips (regresión jerárquica por day).
- McElreath's Howell1 o chimpanzees (modelos clásicos).
- Librerías: pymc (≥ 5), numpyro, arviz, jax.

Ejercicios

1. PyMC v5 hierarchical: tip ~ Normal(α_day + β·bill, σ); α_day ~ Normal(μα, σα).
2. Non-centered: re-parametrizar el modelo con α_day = μα + σα · z_day, z_day ~ N(0,1). Comparar divergences.
3. PPC: pm.sample_posterior_predictive + az.plot_ppc. Decidir si modelo razonable.
4. NumPyro version: traducir, comparar tiempo.
5. LOO compare: 3 modelos (intercepto solo, + slope, + jerárquico). az.compare.

Homework verificable

Modelo bayesiano completo sobre tips:

1. PyMC v5 jerárquico con day como nivel.
2. NumPyro mismo modelo.
3. SVI en NumPyro como alternativa rápida.
4. Comparar 3 enfoques: MCMC PyMC, MCMC NumPyro, SVI NumPyro.
5. az.plot_trace, az.summary, az.compare para 2 variantes del modelo.

Criterio de aceptación: convergencia (r_hat ≤ 1.01); SVI cierra gap con MCMC en < 30 s; ArviZ plots producidos.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Divergences > 100	Funnel posterior. Fix: non-centered parame
r_hat = 1.3	No convergió. Fix: más tune, target_accept
Prior Uniform(-1e6, 1e6)	Plano no es "no informativo". Fix: weakly
SVI con resultados raros	Posterior multimodal o AutoNormal no aplic
Comparar modelos con DIC	Deprecated. Fix: LOO o WAIC.

Preguntas frecuentes

PyMC v5 o NumPyro?

PyMC v5 si tu modelo ya está en PyMC3/v4 (migración fácil). NumPyro si necesitás velocidad o GPU/TPU.

SVI o MCMC?

MCMC para inferencia rigurosa. SVI para producción donde latencia importa.

Stan?

Sí, alternativa madura. cmdstanpy. Sintaxis Stan propia. ArviZ integra.

Edward2 / TFP?

TensorFlow Probability. Menos comunidad que PyMC/NumPyro. Bueno si ya en TF.

Modelos jerárquicos cuándo non-centered?

Casi siempre. Centered solo si hay mucho data por nivel.

Referencias

- Salvatier, Wiecki & Fonnesbeck (2016), PyMC3 — original paper.
- Phan, Pradhan & Jankowiak (2019), NumPyro.
- Vehtari et al. (2017), Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC.
- McElreath (2020), Statistical Rethinking — best book intro bayesiano.

Siguiente parte

Clase 194 — Versionado de datos con DVC

Apéndice: notebook (primer bloque)

PyMC v5 (PyTensor backend), NumPyro (JAX, NUTS rápido), ArviZ (diagnósticos y comparación de modelos). Requiere: pip install numpy scipy arviz (opcional pymc, numpyro).

```
import numpy as np
rng = np.random.default_rng(42)
n = 200
TRUE_A, TRUE_B, TRUE_SIGMA = 1.5, 2.3, 1.0
x = rng.normal(0, 1, n)
y = TRUE_A + TRUE_B * x + rng.normal(0, TRUE_SIGMA, n)
print(f'n={n}  truth: a={TRUE_A}  b={TRUE_B}  sigma={TRUE_SIGMA}')
```

Archivos complementarios

- notebook.ipynb