

MARCO DE ANÁLISIS DE SENSIBILIDAD PARA MODELOS BAYESIANOS DE DESAGREGACIÓN ECONÓMICA: UN MANUAL INTEGRAL

José Mauricio Gómez Julián

2025-09-16

Contents

1. Planteamiento del problema	2
2. Construcción del likelihood sectorial L	3
2.1 PCA/SVD de la matriz prior centrada	3
2.2 Expansión temporal de L	3
3. Reglas de actualización del posterior (deterministas, sin MCMC)	4
4. Coherencia, estabilidad e interpretabilidad	4
4.1 Coherencia respecto de L	4
4.2 Estabilidad numérica y temporal	5
4.3 Interpretabilidad	5
5. API de extremo a extremo (bayesian_disaggregate)	6
6. Cómo interpretar las visualizaciones clave	6
7. Demo sintética reproducible (se ejecuta al compilar)	6
8. Pipeline con datos reales (deshabilitado por defecto)	7
9. Guía práctica y defaults	10
Apéndice A. Invariantes y chequeos rápidos	10

Cómo leer este manual.

Las Secciones 1–3 presentan la **teoría** (con ecuaciones); las Secciones 4–6 introducen **métricas y diagnósticos**; las Secciones 7–8 ofrecen **código reproducible**: un demo sintético (se ejecuta al compilar) y un **pipeline con datos reales** (deshabilitado por defecto para compilar rápido). Todo el código está alineado con el paquete **BayesianDisaggregation** (funciones, defaults, diagnósticos).

```
# Ajustes globales de knitr
knitr::opts_chunk$set(
  echo = TRUE, message = FALSE, warning = FALSE,
  fig.width = 9, fig.height = 6
)

# Paquetes
suppressPackageStartupMessages({
  library(BayesianDisaggregation) # Paquete desarrollado
  library(dplyr)
  library(tidyr)
  library(ggplot2)
  library(readr)
  library(openxlsx)
})
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
# Verbosidad del logger del paquete
log_enable("INFO")
set.seed(2024)
```

1. Planteamiento del problema

Observamos un **índice agregado** (p.ej., CPI) por periodo $t = 1, \dots, T$ y queremos una **desagregación sectorial** en K componentes con participaciones por periodo sobre el **símplice unitario**:

$$W_t = (w_{t1}, \dots, w_{tK}), \quad w_{tk} \geq 0, \quad \sum_{k=1}^K w_{tk} = 1.$$

Partimos de una **matriz prior** $P \in \mathbb{R}^{T \times K}$ (filas en el símplice) y construimos un **likelihood sectorial** $L \in \Delta^{K-1}$ (vector no negativo que suma 1). Un **perfil temporal** lo expande a $LT \in \mathbb{R}^{T \times K}$. Por último, una **regla determinista de actualización** combina P y LT para obtener el posterior W .

2. Construcción del likelihood sectorial L

2.1 PCA/SVD de la matriz prior centrada

Sea P validada (finita, no negativa, filas ≈ 1 ; pequeñas desviaciones se renormalizan). **Centramos** columnas en el tiempo:

$$X = P - \mathbf{1}\bar{p}^\top, \quad \bar{p} = \frac{1}{T} \sum_{t=1}^T P_t.$$

Calculamos SVD $X = U\Sigma V^\top$. Sea v el **primer vector de carga** (PC1). Mapeamos a saliencia no negativa y normalizamos:

$$\ell_k = |v_k|, \quad L_k = \frac{\ell_k}{\sum_j \ell_j}.$$

Si PC1 es **degenerada** (varianza casi nula, columnas casi idénticas), usamos **medias de columna** de P (renormalizadas). Implementado por:

```
# Ejemplo de llamada (la implementación está en el paquete):  
# L <- compute_L_from_P(P)
```

Diagnóstico adjunto a L: atributos "pc1_loadings", "explained_var" y "fallback".

2.2 Expansión temporal de L

Creamos LT aplicando un perfil temporal $w_t \geq 0$ y renormalizando por fila:

$$LT_{t,k} \propto w_t L_k, \quad \sum_k LT_{t,k} = 1.$$

Patrones **incluidos**:

- **constant:** $w_t = 1$
- **recent:** peso lineal creciente en t
- **linear:** rampa afín entre extremos
- **bell:** campana (tipo Gauss) simétrica alrededor de $T/2$

```
# Ejemplo:  
# LT <- spread_likelihood(L, T_periods = nrow(P), pattern = "recent")
```

3. Reglas de actualización del posterior (deterministas, sin MCMC)

Dados P y LT (ambos por fila en el símplice), definimos cuatro reglas:

- **Promedio ponderado** ($\lambda \in [0, 1]$):

$$W = \text{norm}_1\{\lambda P + (1 - \lambda)LT\}.$$

- **Multiplicativa** (producto elemento a elemento y renormalización):

$$W = \text{norm}_1\{P \odot LT\}.$$

- **Media Dirichlet** (conjugación analítica, $\gamma > 0$; γ menor → más concentración):

$$\alpha_{\text{post}} = \frac{P}{\gamma} + \frac{LT}{\gamma}, \quad W = \frac{\alpha_{\text{post}}}{\mathbf{1}^\top \alpha_{\text{post}}}.$$

- **Adaptativa** (mezcla sectorial según volatilidad histórica del prior):

$$\phi_k = \min\left(\frac{\sigma_k}{\bar{\sigma}}, 0.8\right), \quad W_{t.} = \text{norm}_1\{(1 - \phi) \odot P_{t.} + \phi \odot LT_{t.}\}.$$

Funciones del paquete:

```
# posterior_weighted(P, LT, lambda = 0.7)
# posterior_multiplicative(P, LT)
# posterior_dirichlet(P, LT, gamma = 0.1)
# posterior_adaptive(P, LT)
```

4. Coherencia, estabilidad e interpretabilidad

4.1 Coherencia respecto de L

Promedios temporales:

$$\bar{p} = \frac{1}{T} \sum_t P_{t.}, \quad \bar{w} = \frac{1}{T} \sum_t W_{t.}.$$

Con $\rho(\cdot, \cdot)$ **correlación robusta** (máximo entre |Pearson| y |Spearman|), la **coherencia** escala el **incremento** $\Delta\rho = \max(0, \rho(\bar{w}, L) - \rho(\bar{p}, L))$:

$$\text{coherence} = \min\{1, \text{const} + \text{mult} \cdot \Delta\rho\}.$$

4.2 Estabilidad numérica y temporal

- **Estabilidad numérica (penalización exponencial)** sobre desvío de suma por fila y negativos:

$$S_{\text{num}} = \exp\{-a \cdot |\sum_k W_{tk} - 1| - b \cdot \#(W < 0)\}.$$

- **Estabilidad temporal** via $|\Delta|$ promedio (menor variación mayor score):

$$S_{\text{tmp}} = \frac{1}{1 + \kappa \cdot |\overline{\Delta W}|}, \quad |\overline{\Delta W}| = \frac{1}{K} \sum_k \frac{1}{T-1} \sum_t |W_{t+1,k} - W_{t,k}|.$$

- **Estabilidad compuesta** (60% numérica, 40% temporal por defecto):

$$S_{\text{comp}} = 0.6 S_{\text{num}} + 0.4 S_{\text{tmp}}.$$

Funciones del paquete:

```
# coherence_score(P, W, L, mult = 3.0, const = 0.5)
# numerical_stability_exp(W, a = 1000, b = 10)
# temporal_stability(W, kappa = 50)
# stability_composite(W, a = 1000, b = 10, kappa = 50)
```

4.3 Interpretabilidad

Dos principios:

1. **Preservación** de estructura sectorial (correlación entre \bar{p} y \bar{w});
2. **Plausibilidad** de cambios promedio sectoriales (penalizar cambios relativos extremos).

Implementación:

$$\text{pres} = \max\{0, \rho(\bar{p}, \bar{w})\}, \quad r_k = \frac{|\bar{w}_k - \bar{p}_k|}{\bar{p}_k + \epsilon}, \quad \text{plaus} = \frac{1}{1 + 2 \cdot Q_{0.9}(r_k)}.$$

Entonces $\text{interp} = 0.6 \text{pres} + 0.4 \text{plaus}$.

```
# interpretability_score(P, W, use_q90 = TRUE)
```

5. API de extremo a extremo (bayesian_disaggregate)

Pipeline conveniente:

1. `read_cpi()` y `read_weights_matrix()` (Excel)
2. `compute_L_from_P(P)` y `spread_likelihood(L, T, pattern)`
3. regla de posterior (`weighted` / `multiplicative` / `dirichlet` / `adaptive`)
4. métricas: coherencia, estabilidad compuesta, interpretabilidad, eficiencia (heurística) y **score compuesto**
5. exportadores: `save_results()` y workbook “todo en uno” para la mejor config.

```
# Firma (ver Sección 8 para datos reales):
# bayesian_disaggregate(path_cpi, path_weights,
#   method = c("weighted", "multiplicative", "dirichlet", "adaptive"),
#   lambda = 0.7, gamma = 0.1,
#   coh_mult = 3.0, coh_const = 0.5,
#   stab_a = 1000, stab_b = 10, stab_kappa = 50,
#   likelihood_pattern = "recent")
```

6. Cómo interpretar las visualizaciones clave

- **Heatmap del posterior W** : cada **celda** es la participación sectorial en un año; **filas** son años, **columnas** son sectores. *Interpretación*: colores más intensos = mayor share; **suavidad horizontal** indica estabilidad temporal; **bandas verticales** reflejan sectores persistentes.
- **Serie de sectores top**: para los sectores de mayor participación promedio, **líneas** con la trayectoria del share. *Interpretación*: niveles estables = estabilidad; cambios de tendencia sugieren shifts macro/estructurales.
- **Hoja Sector_CPI**: $\hat{Y}_{t,k} = \text{CPI}_t \times W_{t,k}$. *Interpretación*: descomposición del agregado en magnitudes sectoriales (escala del índice).

7. Demo sintética reproducible (se ejecuta al compilar)

Pequeño ejemplo seguro para compilar.

```
# Prior sintético (filas en símlice)
T <- 10; K <- 6
set.seed(123)
P <- matrix(rexp(T*K), nrow = T)
P <- P / rowSums(P)

# Likelihood a partir de P (PCA/SVD; robusto con fallback)
L <- compute_L_from_P(P)
```

```

# Expansión temporal "recent"
LT <- spread_likelihood(L, T_periods = T, pattern = "recent")

# Dos posteriors
W_weighted <- posterior_weighted(P, LT, lambda = 0.7)
W_adaptive <- posterior_adaptive(P, LT)

# Métricas para el adaptativo
coh <- coherence_score(P, W_adaptive, L)
stab <- stability_composite(W_adaptive, a = 1000, b = 10, kappa = 50)
intr <- interpretability_score(P, W_adaptive)
eff <- 0.65
comp <- 0.30*coh + 0.25*stab + 0.25*intr + 0.20*eff

data.frame(coherence = coh, stability = stab, interpretability = intr,
           efficiency = eff, composite = comp) %>% round(4)

```

```

##      coherence stability interpretability efficiency composite
## 90%          1      0.7537           0.6887          0.65      0.7906

```

8. Pipeline con datos reales (deshabilitado por defecto)

Poné `eval=TRUE` después de configurar tus rutas. Por defecto queda en `FALSE` para compilar en cualquier máquina.

```

# === Rutas (usar / en Windows) ===
path_cpi <- "E:/Carpeta de Estudio/[Teoría Marxista]/6. [Mis Investigaciones]/ANÁLISIS DINÁMICO"
path_w <- "E:/Carpeta de Estudio/[Teoría Marxista]/6. [Mis Investigaciones]/ANÁLISIS DINÁMICO"
out_dir <- "E:/Carpeta de Estudio/[Teoría Marxista]/6. [Mis Investigaciones]/ANÁLISIS DINÁMICO"
if (!dir.exists(out_dir)) dir.create(out_dir, recursive = TRUE)

# --- Corrida base (default robustos) ---
base_res <- bayesian_disaggregate(
  path_cpi      = path_cpi,
  path_weights  = path_w,
  method       = "adaptive",
  lambda        = 0.7,    # registrado en métricas; no usado por "adaptive"
  gamma        = 0.1,
  coh_mult     = 3.0,
  coh_const    = 0.5,
  stab_a       = 1000,
  stab_b       = 10,
  stab_kappa   = 60,
  likelihood_pattern = "recent"
)

```

```

xlsx_base <- save_results(base_res, out_dir = file.path(out_dir, "base"))
print(base_res$metrics)

# --- Búsqueda en malla paralela (compacta y discriminativa) ---
n_cores <- max(1, parallel::detectCores() - 4)
grid_df <- expand.grid(
  method      = c("weighted", "multiplicative", "dirichlet", "adaptive"),
  lambda      = c(0.5, 0.7, 0.9), # sólo para "weighted"
  gamma       = c(0.05, 0.1, 0.2), # sólo para "dirichlet"
  coh_mult    = c(2.5, 3.0, 3.5),
  coh_const   = c(0.4, 0.5, 0.6),
  stab_a      = 1000,
  stab_b      = 10,
  stab_kappa  = c(40, 60, 80),
  likelihood_pattern = c("recent", "bell"),
  KEEP.OUT.ATTRS = FALSE,
  stringsAsFactors = FALSE
)

grid_res <- run_grid_search(
  path_cpi    = path_cpi,
  path_weights = path_w,
  grid_df     = grid_df,
  n_cores     = n_cores
)

write.csv(grid_res, file.path(out_dir, "grid_results.csv"), row.names = FALSE)

best_row <- grid_res %>% arrange(desc(composite)) %>% slice(1)
print(best_row)

# --- Re-ejecutar la mejor configuración para exportar limpio ---
best_res <- bayesian_disaggregate(
  path_cpi      = path_cpi,
  path_weights  = path_w,
  method        = best_row$method,
  lambda        = if (!is.na(best_row$lambda)) best_row$lambda else 0.7,
  gamma         = if (!is.na(best_row$gamma)) best_row$gamma else 0.1,
  coh_mult      = best_row$coh_mult,
  coh_const     = best_row$coh_const,
  stab_a        = best_row$stab_a,
  stab_b        = best_row$stab_b,
  stab_kappa    = best_row$stab_kappa,
  likelihood_pattern = best_row$likelihood_pattern
)

xlsx_best <- save_results(best_res, out_dir = file.path(out_dir, "best"))

# --- Un Excel con todo (hiperparámetros incluidos) ---

```



```

sector_summary <- tibble(
  Sector      = colnames(best_res$posterior)[-1],
  prior_mean  = colMeans(as.matrix(best_res$prior[, -1])),
  posterior_mean = colMeans(as.matrix(best_res$posterior[, -1]))
)

wb <- createWorkbook()
addWorksheet(wb, "Hyperparameters"); writeData(wb, "Hyperparameters", best_row)
addWorksheet(wb, "Metrics"); writeData(wb, "Metrics", best_res$metrics)
addWorksheet(wb, "Prior_P"); writeData(wb, "Prior_P", best_res$prior)
addWorksheet(wb, "Posterior_W"); writeData(wb, "Posterior_W", best_res$posterior)
addWorksheet(wb, "Likelihood_t"); writeData(wb, "Likelihood_t", best_res$likelihood_t)
addWorksheet(wb, "Likelihood_L"); writeData(wb, "Likelihood_L", best_res$likelihood)

addWorksheet(wb, "Sector_Summary"); writeData(wb, "Sector_Summary", sector_summary)

for (sh in c("Hyperparameters", "Metrics", "Prior_P", "Posterior_W",
             "Likelihood_t", "Likelihood_L", "Sector_Summary")) {
  freezePane(wb, sh, firstRow = TRUE)
  addFilter(wb, sh, rows = 1, cols = 1:ncol(readWorkbook(wb, sh)))
  setColWidths(wb, sh, cols = 1:200, widths = "auto")
}

# --- Agregar CPI sectorial:  $\hat{Y}_{t,k} = CPI_t * W_{t,k}$  ---
W_post <- best_res$posterior # Year + sectores
cpi_df <- read_cpi(path_cpi) # Year, CPI

sector_cpi <- dplyr::left_join(W_post, cpi_df, by = "Year") %>%
  dplyr::mutate(dplyr::across(-c(Year, CPI), ~ .x * CPI))

# Chequeo de suma vs CPI
check_sum <- sector_cpi %>%
  dplyr::mutate(row_sum = rowSums(dplyr::across(-c(Year, CPI))),
               diff = CPI - row_sum)
print(head(check_sum, 5))

addWorksheet(wb, "Sector_CPI")
writeData(wb, "Sector_CPI", sector_cpi)
freezePane(wb, "Sector_CPI", firstRow = TRUE)
addFilter(wb, "Sector_CPI", rows = 1, cols = 1:ncol(sector_cpi))
setColWidths(wb, "Sector_CPI", cols = 1:200, widths = "auto")

excel_onefile <- file.path(out_dir, "best", "Best_Full_Output_withSectorCPI.xlsx")
saveWorkbook(wb, excel_onefile, overwrite = TRUE)

# --- Gráficos rápidos (PNG) ---
dir_plots <- file.path(out_dir, "best", "plots")

```

```

if (!dir.exists(dir_plots)) dir.create(dir_plots, recursive = TRUE)

W_long <- best_res$posterior %>%
  pivot_longer(-Year, names_to = "Sector", values_to = "Weight")
p_heat <- ggplot(W_long, aes(Year, Sector, fill = Weight)) +
  geom_tile() + scale_fill_viridis_c() +
  labs(title = "Posterior weights (W): heatmap", x = "Año", y = "Sector", fill = "Share") +
  theme_minimal(base_size = 11) + theme(axis.text.y = element_text(size = 6))
ggsave(file.path(dir_plots, "posterior_heatmap.png"), p_heat, width = 12, height = 9, dpi = 220)

top_sectors <- best_res$posterior %>%
  summarise(across(-Year, mean)) %>%
  pivot_longer(everything(), names_to = "Sector", values_to = "MeanShare") %>%
  arrange(desc(MeanShare)) %>% slice(1:8) %>% pull(Sector)

p_lines <- best_res$posterior %>%
  select(Year, all_of(top_sectors)) %>%
  pivot_longer(-Year, names_to = "Sector", values_to = "Weight") %>%
  ggplot(aes(Year, Weight, color = Sector)) +
  geom_line(linewidth = 0.9) +
  labs(title = "Top 8 sectores por share promedio (posterior W)", y = "Share", x = "Año") +
  theme_minimal(base_size = 11)
ggsave(file.path(dir_plots, "posterior_topSectors.png"), p_lines, width = 11, height = 6, dpi = 220)

```

9. Guía práctica y defaults

- Usá `method="adaptive"` cuando la **volatilidad sectorial** del prior sea heterogénea; si no, `weighted` con $\lambda \in [0.7, 0.9]$ suele rendir muy bien y frecuentemente lidera la grilla.
- Los parámetros de **coherencia** por defecto (`mult=3.0`, `const=0.5`) brindan un score 0–1 interpretable que enfatiza la **mejora** frente al prior.
- La penalización **exponencial** en estabilidad numérica es deliberadamente estricta: evita desviaciones de suma y negativos en corridas automatizadas y búsquedas en grilla.
- Para informes, exportá **Sector_CPI** y mostrás la descomposición $\hat{Y}_{t,k}$.

Apéndice A. Invariantes y chequeos rápidos

```

# Invariantes sobre un ejemplo sintético
T <- 6; K <- 5
set.seed(7)
P <- matrix(rexp(T*K), nrow = T); P <- P / rowSums(P)
L <- compute_L_from_P(P)
LT <- spread_likelihood(L, T, "recent")
W <- posterior_multiplicative(P, LT)

```

```
# Invariantes
stopifnot(all(abs(rowSums(P) - 1) < 1e-12))
stopifnot(all(abs(rowSums(LT) - 1) < 1e-12))
stopifnot(all(abs(rowSums(W) - 1) < 1e-12))
c(
  coherence = coherence_score(P, W, L),
  stability = stability_composite(W),
  interpret = interpretability_score(P, W)
) %>% round(4)
```

```
##      coherence      stability interpret.90%
##      1.0000      0.6459      0.6245
```

Apéndice B. Información de la sesión

```
sessionInfo()
```

```
## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.utf8  LC_CTYPE=Spanish_Spain.utf8
## [3] LC_MONETARY=Spanish_Spain.utf8 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.utf8
##
## time zone: America/Costa_Rica
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] openxlsx_4.2.8      readr_2.1.5
## [3] ggplot2_4.0.0       tidyr_1.3.1
## [5] dplyr_1.1.4         BayesianDisaggregation_0.1.0
##
## loaded via a namespace (and not attached):
## [1] gtable_0.3.6      compiler_4.4.2    tidyselect_1.2.1  Rcpp_1.1.0
## [5] tinytex_0.57      zip_2.3.3         scales_1.4.0      yaml_2.3.10
## [9] fastmap_1.2.0     R6_2.6.1          generics_0.1.4    knitr_1.50
```

## [13]	iterators_1.0.14	tibble_3.3.0	tzdb_0.5.0	pillar_1.11.0
## [17]	RColorBrewer_1.1-3	rlang_1.1.5	stringi_1.8.7	xfun_0.53
## [21]	S7_0.2.0	cli_3.6.3	withr_3.0.2	magrittr_2.0.4
## [25]	digest_0.6.37	foreach_1.5.2	grid_4.4.2	rstudioapi_0.17.1
## [29]	hms_1.1.3	lifecycle_1.0.4	vctrs_0.6.5	evaluate_1.0.5
## [33]	glue_1.8.0	farver_2.1.2	codetools_0.2-20	rmarkdown_2.29
## [37]	purrr_1.1.0	tools_4.4.2	pkgconfig_2.0.3	htmltools_0.5.8.1