SENSITIVITY ANALYSIS FRAMEWORK FOR BAYESIAN ECONOMIC DISAGGREGATION: A COMPREHENSIVE MANUAL

José Mauricio Gómez Julián

2025-09-16

Contents

How to read this manual.
Sections 1–3 develop the **theory** (with equations); Sections 4–6 give **diagnostics** and **metrics**; Sections 7–8 provide **reproducible code**: a fast synthetic demo (evaluates on knit) and a full **real-data pipeline** (disabled by default for speed, enable by setting `eval=TRUE`). All code is consistent with the functions exported by the `BayesianDisaggregation` package.

```
# Global chunk defaults
knitr::opts_chunk$set(
  echo = TRUE, message = FALSE, warning = FALSE,
  fig.width = 9, fig.height = 6
)

# Libraries
suppressPackageStartupMessages({
  library(BayesianDisaggregation)
  library(dplyr)
  library(tidyr)
  library(ggplot2)
  library(readr)
  library(openxlsx)
})
```

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

```
# Logging verbosity from the package
log_enable("INFO")
set.seed(2024)
```

1. Problem Setup

We observe an **aggregate index** (e.g., CPI) by period $t = 1, \dots, T$, and we want a **sectoral disaggregation** into K components whose period-wise shares lie on the **unit simplex**:

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

We start with a **prior weight matrix** $P \in \mathbb{R}^{T \times K}$ (rows on the simplex), and construct a **likelihood of sectors** $L \in \Delta^{K-1}$ (a non-negative vector summing to one). A **temporal profile** then spreads L to $LT \in \mathbb{R}^{T \times K}$. Finally, a **deterministic update rule** combines P and LT to obtain the posterior W.

2. Constructing the Sectoral Likelihood L

2.1 PCA/SVD of the centered prior matrix

Let P be validated (finite, non-negative, rows ≈ 1; small deviations renormalized). We **center** columns over time:

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

Compute the SVD $X = U\Sigma V^{\top}$. Let v denote the **first right singular vector** (PC1 loadings). We map to non-negative salience via absolute values and normalize:

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

If PC1 is **degenerate** (near-zero variance or identical columns), we fall back to **column means** of P (renormalized). This is implemented in:

```
# Example call (internals are in the package):
# L <- compute_L_from_P(P)
```

Diagnostics attached to `L`: attributes `"pc1_loadings"`, `"explained_var"`, and `"fallback"`.

2.2 Temporal spreading of L

We create a time-varying matrix LT by applying a non-negative weight profile w_t and row-renormalizing:

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

Built-in **patterns**:

- `constant`: $w_t = 1$
- `recent`: linearly increasing in t (more weight to recent periods)
- `linear`: affine ramp between endpoints
- `bell`: symmetric Gaussian-like bump around $T/2$

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

3. Posterior Updating Rules (Deterministic, MCMC-free)

Given P and LT (both row-wise on the simplex), we define four deterministic updates:

- **Weighted average** (mixing parameter $\lambda \in [0, 1]$):

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

- **Multiplicative** (elementwise product with re-normalization):

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

- **Dirichlet mean** (analytical conjugacy, $\gamma > 0$, smaller $\gamma \Rightarrow$ sharper):

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

- **Adaptive** (sector-wise mixing by prior volatility):

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

All are exposed in the package:

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

4. Coherence, Stability, and Interpretability

4.1 Coherence with respect to L

Define prior/posterior **temporal means**:

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

Let $\rho(\cdot, \cdot)$ be a **robust correlation** (max of |Pearson| and |Spearman|). The **coherence** scales the **increment** $\Delta\rho = \max(0, \rho(\bar{w}, L) - \rho(\bar{p}, L))$:

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

4.2 Numerical and temporal stability

- **Numerical stability (exponential penalty)** on row-sum deviation and negatives:

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

- **Temporal stability** via average $|\Delta|$ (lower variation $\Rightarrow$ higher score):

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

- **Composite stability** (default weights 60% numeric, 40% temporal):

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

The package functions:

```
# 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 Interpretability

Two principles:

1. **Preservation** of the sectoral structure (correlation between $\bar{p}$ and $\bar{w}$);
2. **Plausibility** of average sector changes (penalize extreme relative shifts).

Implementation:

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

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

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

5. End-to-End API (`bayesian_disaggregate`)

The convenience pipeline:

1. `read_cpi()` and `read_weights_matrix()` (Excel)
2. `compute_L_from_P(P)` and `spread_likelihood(L, T, pattern)`
3. posterior rule (`weighted` / `multiplicative` / `dirichlet` / `adaptive`)
4. metrics: coherence, stability (composite), interpretability, efficiency (heuristic), composite score
5. export helpers: `save_results()` and a one-file workbook for "best" config

```
# Example signature (see Section 8 for real data):
# 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. Interpreting Key Visualizations

- **Heatmap of posterior** W: each **cell** is a sector share in a year; **rows** are years, **columns** are sectors. *Read it as*: darker tiles = larger sector share; **horizontal smoothness** indicates temporal stability; **vertical patterns** (bands) show persistent sectoral importance.

- **Top-sectors lines**: for the most relevant sectors by average share, **lines** track the sector's share over time. *Read it as*: consistent levels = stability; trend changes coincide with macro structure shifts.

- **Sectoral CPI sheet**: $\hat{Y}_{t,k} = \mathrm{CPI}_t \times W_{t,k}$. *Read it as*: dollarized (or index-scaled) decomposition of the aggregate.

7. Reproducible Synthetic Demo (evaluates on knit)

This chunk synthesizes a small example you can knit safely.

```
# Synthetic prior matrix (rows on simplex)
T <- 10; K <- 6
set.seed(123)
P <- matrix(rexp(T*K), nrow = T)
P <- P / rowSums(P)

# Likelihood vector from P (PCA/SVD; robust with fallback)
L  <- compute_L_from_P(P)
```

```r
# Spread over time with "recent" pattern
LT <- spread_likelihood(L, T_periods = T, pattern = "recent")

# Try a couple of posteriors
W_weighted <- posterior_weighted(P, LT, lambda = 0.7)
W_adaptive <- posterior_adaptive(P, LT)

# Metrics for adaptive
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. Full Real-Data Pipeline (disable/enable evaluation)

> **Switch to `eval=TRUE` after setting your paths**. By default we keep this chunk off to render quickly on any machine.

```r
# === Paths (use forward slashes on 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)

# --- Base run (robust defaults) ---
base_res <- bayesian_disaggregate(
  path_cpi          = path_cpi,
  path_weights      = path_w,
  method            = "adaptive",
  lambda            = 0.7,   # recorded in metrics; not used by "adaptive"
  gamma             = 0.1,
  coh_mult          = 3.0,
  coh_const         = 0.5,
  stab_a            = 1000,
  stab_b            = 10,
  stab_kappa        = 60,
  likelihood_pattern = "recent"
)
```

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

# --- Parallel grid search (compact yet discriminative) ---
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),    # only used if method == "weighted"
  gamma              = c(0.05, 0.1, 0.2),   # only used if method == "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-run the best configuration for clean export ---
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"))

# --- One Excel with everything (including hyperparameters) ---
```

```r
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")
}

# --- Add sectoral CPI (aggregate times posterior weights) ---
W_post <- best_res$posterior            # Year + sectors
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))

# Quality check: sector sums 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)

# --- Quick plots (saved as PNGs) ---
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 = "Year", 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 = 22

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 sectors by average share (posterior W)", y = "Share", x = "Year") +
  theme_minimal(base_size = 11)
ggsave(file.path(dir_plots, "posterior_topSectors.png"), p_lines, width = 11, height = 6, dpi =
```

9. Practical Guidance and Defaults

- Prefer `method="adaptive"` when prior sector volatilities are heterogeneous; otherwise `weighted` with $\lambda \in [0.7, 0.9]$ is strong and often tops the grid.
- The default **coherence** parameters (`mult=3.0, const=0.5`) produce a bounded, interpretable 0–1 score that emphasizes **improvement** over the prior.
- The **exponential** numerical penalty is intentionally sharp: it keeps row-sum deviations and negatives at bay in automated runs and grid searches.
- For reports, export **Sector_CPI** to illustrate the economic decomposition $\hat{Y}_{t,k}$.

Appendix A. Invariants and Quick Checks

```
# Example: invariants on a fresh synthetic run
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)

# Invariants
```

```r
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
```

Appendix B. Session Info

```r
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
```

Notes

- The *real-data* chunk is set to `eval=FALSE` so the manual renders anywhere. Flip it to `TRUE` on your machine to run fully against your Excel files.
- The "best one-file" export includes **Hyperparameters, Metrics, Prior_P, Posterior_W, Likelihood_t, Likelihood_L, Sector_Summary, Sector_CPI**, with frozen headers and filters for quick analysis.
- Plots are written to `.../best/plots/` and match the interpretation guidance in Section 6.