

Variational Bayesian Inference.

Mixture Models with a view to Dynamic Models.

Javier López-de-Lacalle

September 2026

Abstract

This document is a version of the Final Degree Project submitted as part of the University Specialisation Degree in *Applied Artificial Intelligence and Its Mathematical Foundations*¹ at the University of the Basque Country.

This project studies variational Bayesian inference from a mainly pedagogical perspective, using Gaussian mixture models as the setting where theory, derivation, implementation, and empirical experimentation are connected. Rather than relying primarily on existing software, the algorithms were implemented almost entirely from scratch in R, including EM, variational EM, CAVI, amortized responsibility mappings, and extensions toward autoregressive mixture models. This implementation-oriented effort required substantial development and debugging work, but it made the connection between the ELBO, the coordinate updates, and the numerical behavior of the algorithms especially transparent. Simulation experiments, although limited to relatively simple data-generating processes, gave encouraging results and supported further development of the implementation. The project concludes with an application to weather-image clustering, where a Bayesian Gaussian mixture model fitted to PCA-reduced RGB histogram features reveal interpretable colour-intensity patterns. The project lays useful foundations for future extensions toward structured variational inference and dynamical variational autoencoders.

¹<https://www.ehu.eus/en/web/graduondokoak/university-specialisation-applied-artificial-intelligence-mathematical-foundations>.

Contents

1	Introduction	4
1.1	Motivation and starting point	4
1.2	Guide to the project	4
1.3	GitHub repository	6
2	Literature review	6
2.1	Amortized variational inference: When and why?	6
2.2	Variational inference: A review for statisticians	7
2.3	Variational Bayes for generalized autoregressive models	8
3	Amortized variational EM	9
3.1	Linear and quadratic ideal inference mappings	11
3.2	Relation with auto-encoding variational Bayes	12
4	Simulation experiments	12
4.1	The classical vs Bayesian approaches	12
4.2	Amortized inference	13
5	Application: Weather images clustering	14
6	Conclusions and future work	15
	References	17
	Appendix	18
A	EM algorithm. Full covariance matrix model.	18
A.1	The Model.	18
A.2	Complete-data likelihood.	19
A.3	EM objective function.	20
A.4	Expectation step.	20
A.5	Maximization step.	20
B	Amortized Variational EM algorithm	21
B.1	Variational distribution for the allocations	21
B.2	Evidence lower bound	22
B.3	Variational E-step	23
B.4	Variational M-step	23

C	Amortized variational inference for the responsibilities	24
C.1	Amortization of the local variational parameters	24
C.2	Softmax parameterization	25
C.3	Neural-network inference mapping	25
C.4	Amortized variational E-step	26
C.5	Initialization	26
D	CAVI algorithm. Normal-Wishart model.	27
D.1	Description of the model	27
D.2	Prior densities	27
D.3	Expression of the log-joint density	28
D.4	Expression of the joint log-likelihood	28
D.5	Variational family	29
D.6	Expected log-density under q , including constants	29
D.7	Entropy terms, including constants	30
D.8	Coordinate-ascent algorithm	32
E	Generalized vector autoregression model.	36
E.1	The model and notation.	36
E.2	Complete-data log-likelihood function.	37
E.3	Priors.	38
E.4	Variational densities.	39
E.5	ELBO.	40
E.6	Coordinate-ascent algorithm.	43

1 Introduction

1.1 Motivation and starting point

This project was not initially formulated around a specific empirical problem or a particular dataset. Its main motivation was to study variational inference and broaden my understanding of a family of techniques that is increasingly important in modern probabilistic machine learning. This involved developing familiarity with the associated computational methods.

The starting point was a background in linear Gaussian models and state-space methods. Analytical closed-form expressions are available there and solutions can be implemented through the Kalman filter and smoother. Kalman's original formulation introduced the filtering problem in the context of projections [1]. Classical references such as Harvey [2] and Durbin and Koopman [3] develop these methods further for time series modelling.

A useful bridge toward probabilistic machine learning was provided by the probabilistic treatment of Bayesian filtering and smoothing in Särkkä and Svensson [4]. This perspective, along with the AAIMF lectures, made explicit to me that the Kalman filter is not only an algorithmic recursion, but also an exact Bayesian inference procedure for a particular class of models. From this viewpoint, variational inference appears naturally as a possible extension when exact filtering or exact posterior computation is no longer available.

The classical linear Gaussian case provides exact analytical solutions in restricted settings. Monte Carlo methods extend Bayesian inference to more general models where analytical solutions are not available. This greater generality comes at the cost of increased computational cost. Variational methods offer a complementary strategy: Exact posterior inference is replaced with an optimization problem over a tractable family of distributions. In large-scale applications, this can provide important computational advantages. In the present project, however, dataset size and computational speed were not the main concerns. The goal was instead to gain insight into the method itself: How the variational objective is constructed, how coordinate ascent updates are derived, and how the resulting algorithms behave in simulations and in a concrete image-clustering task.

1.2 Guide to the project

This project consists of four complementary elements: the main text, the appendix, three accompanying Jupyter notebooks and the AAIMF package of R [5] developed for this project.

Main text: The first part of the present document. It can be read independently from beginning to end to get a general overview. It presents the motivation, methodology, the overall results and conclusions without lengthy technical interruptions. The readers can then consult the appendix for the analytical derivations underlying the methods and the notebooks for the computational details, simulations, and image-data application.

Appendix: A substantial part of the project was devoted to working through the analytical details underlying the variational algorithms. For each model, the procedure is broadly similar.

1. Write the complete-data likelihood in terms of latent variables.
2. Specify prior distributions and a factorized variational family.
3. Decompose the evidence lower bound into expected log-likelihood, expected log-prior, and entropy terms.
4. Derive coordinate-ascent updates by analytical derivation on the lower bound with respect to the variational factors.

These analytical derivations are collected in the appendix at the end of this document. It adopts a deliberately concise and schematic style. The derivations are organized as a sequence of definitions, intermediate and resulting expressions. Straightforward transitions such as “hence”, “then”, “we have” followed by mathematical expressions are used broadly. The results therein are also the analytical foundations of the implementation in the **AAIMF** package.

Jupyter notebooks: Three Jupyter notebooks contain two simulation exercises and an application to an image dataset. This format is particularly well suited for these purposes. It combines R code, mathematical expressions, and textual descriptions of the setting and results in an interactive format. Each Jupyter notebook is self-contained and can largely be read and executed independently.

AAIMF package of R: The implementation relies on the analytical expressions given in the appendix and is required in the Jupyter notebooks. The package implements most of the algorithms and models studied in this project. Rather than using mature software tools, this approach was adopted for pedagogical reasons. The additional effort required to design and debug the code provided a better understanding of the methodology. Admittedly, implementing the methods independently entails risks, such as overlooking computational issues or technical details. Nevertheless, for the purposes of this project the risks were outweighed by the benefits.

1.3 GitHub repository

The AAIMF R package, the notebooks and results, as well as other scripts and data required to reproduce the results are available in the project repository: <https://github.com/javlacalle/repository>.

The rest of this document is organised as follows. The literature reviewed is presented in § 2. In § 3 the AVEM algorithm is developed. Simulation experiments are described in Section 4. Section 5 presents an application to clustering a weather-image dataset. Section 6 gives concluding remarks and discusses directions for future work. Technical derivations are given in the Appendix.

2 Literature review

I reviewed the literature that naturally followed the references cited in the Introduction and Bishop [6, Chapters 8–10]. The review gradually led to three central references, each contributing a different perspective on variational inference and its implementation.

2.1 Amortized variational inference: When and why?

The first major reference was the work of Margossian and Blei on amortized variational inference [7]. This paper was useful because it distinguishes between factorized variational inference and amortized variational inference, and studies when an inference function can reproduce the optimum that would otherwise be obtained separately for each observation or dataset.

The linear model described in this paper was worked out in detail and served as a conceptual foundation for further designs discussed in Section 3.

Derivation strategy for the CAVI updates. Appendix A.1 of Margossian and Blei derives the usual coordinate-ascent variational inference rule. Consider the mean-field family

$$q(\mathbf{z}) = \prod_{j=1}^J q_j(z_j), \quad \text{and let} \quad q_{-j}(\mathbf{z}_{-j}) = \prod_{\ell \neq j} q_\ell(z_\ell).$$

When all factors except q_j are held fixed, the terms of the ELBO that depend on q_j can be written as

$$\mathcal{L}(q_j, q_{-j}) = \int q_j(z_j) \mathbb{E}_{q_{-j}} \log p(\mathbf{x}, \mathbf{z}) \, dz_j - \int q_j(z_j) \log q_j(z_j) \, dz_j + C,$$

where C does not depend on q_j . Define the normalized density

$$\tilde{p}_j(z_j) \propto \exp \left\{ \mathbb{E}_{q_{-j}} \log p(\mathbf{x}, \mathbf{z}) \right\}.$$

The coordinate-dependent part of the ELBO is then

$$\mathcal{L}(q_j, q_{-j}) = -\text{KL}(q_j(z_j) \parallel \tilde{p}_j(z_j)) + C',$$

and is maximized when the Kullback–Leibler divergence vanishes. This gives the CAVI rule

$$q_j^*(z_j) = \frac{\exp \left\{ \mathbb{E}_{q_{-j}} [\log p(\mathbf{x}, \mathbf{z})] \right\}}{\int \exp \left\{ \mathbb{E}_{q_{-j}} [\log p(\mathbf{x}, \mathbf{z})] \right\} dz_j},$$

or, equivalently, $\log q_j^*(z_j) = \mathbb{E}_{q_{-j}} \log p(\mathbf{x}, \mathbf{z}) + \text{constant}$.

This rule often provides the shortest derivation of a CAVI update, especially in conditionally conjugate models. It does not, however, remove all the algebraic work. After evaluating the expectation, one must recognize the resulting kernel as that of a familiar distribution and identify its updated parameters. For a Gaussian factor this is usually accomplished by collecting the linear and quadratic terms and completing the square. For other factors, the appropriate distributional form may be less apparent, and the transition from the expected log-joint density to an implementable update can consequently be difficult to spot.

For this reason, the update expressions used in the implementation were derived mainly by parameterizing the variational factors, writing the ELBO explicitly as a function of their parameters, and differentiating it analytically. Thus, for a variational parameter vector $\boldsymbol{\lambda}_j$, the update was obtained from

$$\nabla_{\boldsymbol{\lambda}_j} \mathcal{L}(\boldsymbol{\lambda}_j, \boldsymbol{\lambda}_{-j}) = \mathbf{0},$$

together with normalization and positive-definiteness constraints. Details are given in Appendix D. This route is more laborious because the expectations, entropy terms, derivatives, and constraints must be handled explicitly. Nevertheless, it provides a systematic and transparent way to trace how each term of the ELBO contributes to the update equations and is particularly useful when translating the derivation into code.

Direct differentiation and the CAVI rule optimize the same ELBO and lead to the same updates. The CAVI rule relies on recognizing a probability kernel, whereas direct differentiation amounts to explicit calculus. The latter was chosen because, despite being more tedious, as witnessed in the appendix, it is a broadly applicable strategy and makes the connection between the mathematical objective and its implementation especially visible. When the resulting kernel is recognizable, the CAVI rule provides a shorter path and a useful way to check the differentiated equations.

2.2 Variational inference: A review for statisticians

The second major reference was the review paper by Blei, Kucukelbir, and McAuliffe [8]. This paper was used as the main entry point for the formulation of variational inference

in standard statistical notation. Its mixture-of-Gaussians example was especially important for the project because it provides a relatively simple but nontrivial model with latent discrete assignments and continuous latent parameters.

The project focused on the factorized variational treatment of the mixture-of-Gaussians model. This involved writing the augmented likelihood, specifying the variational family, deriving the evidence lower bound, and obtaining coordinate-ascent variational inference updates. These derivations were later translated into R code, first for simulation experiments and then for image-clustering applications. The details of the expressions involved in the algorithm are given in [Appendix D](#).

2.3 Variational Bayes for generalized autoregressive models

The third major reference was Roberts and Penny [9], which applies variational Bayes to generalized autoregressive models. This paper was important because it connects variational methods with dynamic models that are of my interest.

The model can be understood as an autoregressive model whose innovation distribution is represented as a mixture of Gaussians. This creates a link between time-series modelling and mixture modelling: the autoregressive component captures temporal dependence, while the mixture component allows the innovation distribution to be non-Gaussian.

The implementation in the package **AAIMF** follows the modelling strategy of Roberts and Penny by treating the autoregressive residuals as innovations generated from a finite of Gaussian components. The conditional mean of the observation is first explained by the autoregressive term $X_{\text{lags}}B$. The remaining innovation is assigned probabilistically to one of K Gaussian *regimes*. The implementation follows the Bayesian variational structure, with responsibilities $q(z_n)$, posterior mixture weights, component-specific innovation means and precision matrices updated by coordinate ascent.

At the same time, the implementation extends the original univariate generalized autoregressive setting to a multivariate vector autoregression. Instead of a scalar AR coefficient vector, the model estimates a matrix B collecting the VAR coefficients, which is then converted back into the matrices $A_1, \dots, A_p$. The innovation distribution is also multivariate, so each mixture component has a full precision matrix Λ_k under a Normal-Wishart variational factor. This allows the model to capture not only regime-dependent means and variances of the innovations, but also contemporaneous dependence across the components of the multivariate time series. In this sense, the code can be viewed as a VAR generalization of the Roberts–Penny mixture-noise idea. Details on the derivation of the coordinate-ascent variational inference algorithm (CAVI) for this model are given in [Appendix E](#).

Unfortunately, at this stage it has not been possible to experiment extensively with this implementation, since other parts of the project that were prerequisites for working

this model, occupied most of the development effort so far. Nevertheless, the work carried out shows that it is feasible to implement a variational version of the Roberts and Penny model in the multivariate VAR setting.

Related literature

Survey and tutorial papers: The following references were also reviewed and contributed to my understanding of the field of variational inference: Lappalainen and Miskin (2000) [10], Jordan, Ghahramani, Jaakkola and Saul, Lawrence (1999) [11], Wainwright, Martin and Jordan (2008) [12].

Stochastic optimization: Hoffman, Blei, Wang and Paisley (2013) [13]. Although the topic is not worked in this project, some of the ideas found in this paper are relevant as possible extensions of the baseline tools developed here. Related ideas presented in Robbins and Monro (1951) [14] and Amari (1998) [15] were also explored.

Variational autoencoders: The ideas on amortized variational inference studied in this project can be applied to variational autoencoders. These are generative models where the posterior distribution is parameterized by an inference network. An entry point to these ideas is Kingma and Welling (2014) [16], where variational inference, neural-network parameterization and stochastic optimization are combined in a unified framework.

3 Amortized variational EM

The baseline EM algorithm for the mixture-of-Gaussians model is sketched in Appendix A. The function `avem` in the package `AAIMF` includes an Amortized Variational EM procedure (AVEM), described in Appendix B. It is interesting because it highlights the relation among the exact EM E-step, ELBO maximization, restricted variational inference, and neural-network amortization.

This variation of the standard EM was motivated in part by [7]. Margossian and Blei contrast factorized variational inference, in which a separate variational parameter is estimated for every local latent variable, with amortized variational inference, in which those local parameters are generated by a common inference function. In their notation, the distinction can be represented schematically as

$$q_F(\boldsymbol{\theta}, \mathbf{z}) = q_0(\boldsymbol{\theta}) \prod_{n=1}^N q_n(z_n; \boldsymbol{\nu}_n) \quad \text{and} \quad q_A(\boldsymbol{\theta}, \mathbf{z}) = q_0(\boldsymbol{\theta}) \prod_{n=1}^N q(z_n; f_\phi(x_n)).$$

Factorized variational inference estimates the local parameters $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_N$ separately. Amortized variational inference instead learns a shared function f_ϕ that maps each

observation to the parameters of its corresponding approximate posterior. The cost of learning this mapping can then be shared, or amortized, across observations.

A central question in Margossian and Blei is whether the class of inference functions is sufficiently expressive to reproduce the optimal factorized variational solution. They formulate this as an interpolation problem:

$$f_\phi(x_n) = \boldsymbol{\nu}_n^*, \quad n = 1, \dots, N,$$

where $\boldsymbol{\nu}_n^*$ denotes the optimal local parameter obtained under factorized variational inference. If such an inference function exists and can be learned, then the amortization gap can be closed.

To illustrate this idea, the authors consider the linear probabilistic model

$$p(\theta) \propto 1, \quad z_n \sim \mathcal{N}(0, 1), \quad x_n | z_n, \theta \sim \mathcal{N}(\theta + \tau z_n, \sigma^2),$$

where τ and σ are fixed. For a factorized Gaussian approximation, the optimal variational mean of z_n is

$$\mathbb{E}_{q_n^*} z_n = \frac{\tau}{\sigma^2 + \tau^2} (x_n - \bar{x}),$$

and its optimal variational variance is a constant, say ξ^2 . Therefore, the local variational parameters can be generated by an inference function whose mean is linear in x_n :

$$f_\phi^{(\mu)}(x_n) = \alpha_0 + \alpha_1 x_n, \quad \text{with} \quad \alpha_0 = -\frac{\tau \bar{x}}{\sigma^2 + \tau^2}, \quad \alpha_1 = \frac{\tau}{\sigma^2 + \tau^2}.$$

Together with the constant variance parameter, only three shared variational parameters are needed, regardless of the sample size. This example makes the main amortization principle particularly transparent: A collection of local posterior approximations can be represented by a common, low-dimensional function of the observations.

The AVEM procedure of the **AAIMF** package adopts this same organizing idea. For a Gaussian mixture, the local latent variable is the component indicator z_n , and its approximate posterior is categorical:

$$q_\eta(z_n = k | \mathbf{x}_n) = r_{nk}(\boldsymbol{\eta}).$$

Instead of estimating an unrelated responsibility vector $\mathbf{r}_n$ for every observation, AVEM obtains all responsibilities from a shared inference map:

$$\mathbf{r}_n(\boldsymbol{\eta}) = \text{softmax} \{ \mathbf{u}_\eta(\boldsymbol{\phi}(\mathbf{x}_n)) \}.$$

The network parameters $\boldsymbol{\eta}$ are estimated by maximizing the ELBO while holding the mixture parameters fixed. The resulting responsibilities are subsequently used in the analytical maximization step for the mixing probabilities, component means and precision matrices. Thus, the algorithm alternates between

$$\boldsymbol{\eta}^{(t+1)} = \arg \max_{\boldsymbol{\eta}} \mathcal{L}(q_\eta, \boldsymbol{\theta}^{(t)}) \quad \text{and} \quad \boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{L}(q_{\eta^{(t+1)}}, \boldsymbol{\theta}).$$

This is a natural specialization of the amortization principle to variational EM. The local categorical parameters are replaced by the outputs of a common inference function, while the global mixture parameters are treated as point estimates rather than as random variables with a variational posterior.

The notebook `main-sim-em-diag-cov.ipynb` experiments with this idea through simulations. The approach therein connects with the idea in Margossian and Blei, although the two settings are not identical. Their principal analysis concerns full Bayesian variational inference, whereas the notebook uses maximum-likelihood estimation for the global mixture parameters. Nevertheless, both constructions seek the same question: can one shared function of the observations reproduce the collection of locally optimal variational parameters?

3.1 Linear and quadratic ideal inference mappings

The comparison becomes especially clear in the two-component diagonal-covariance mixture used in the notebook. There, it is shown that conditional on fixed mixture parameters, the exact posterior log-odds satisfy

$$g_{\theta}(\mathbf{x}) = \log \frac{p_{\theta}(z = 2 \mid \mathbf{x})}{p_{\theta}(z = 1 \mid \mathbf{x})} = c + \mathbf{a}^{\top} \mathbf{x} + \mathbf{b}^{\top} (x_1^2, \dots, x_D^2)^{\top}.$$

The exact responsibility of the second component is therefore

$$p_{\theta}(z = 2 \mid \mathbf{x}) = \text{logistic} \{g_{\theta}(\mathbf{x})\}.$$

As a function of the observations, the ideal log-odds mapping is quadratic. This matches the idea of the linear example of Margossian and Blei. In both cases, the locally optimal variational parameters are generated by a common function whose finite-dimensional form can be derived analytically. The main difference is that the mixture model requires a logistic transformation of a quadratic predictor, whereas their illustrative model requires a Gaussian variational factor with a mean linear in the observation and a constant variance.

For the two-dimensional mixture considered in the notebook, the ideal log-odds contain five coefficients: $g_{\theta}(x_1, x_2) = c + a_1 x_1 + a_2 x_2 + b_1 x_1^2 + b_2 x_2^2$. Consequently, the target function is moderately more complicated than in the linear model.

The neural network used in the notebook is more expressive than is necessary for this particular model. This additional flexibility is, however, deliberate and forms part of the purpose of the simulation. It follows a strategy similar to that adopted by Margossian and Blei: The experiment starts from a tractable model for which the exact posterior responsibility mapping is known and can be computed analytically. The neural network is then used to approximate that mapping through ELBO maximization, making it possible to assess whether the amortized procedure recovers a known benchmark. The

objective is therefore not to propose the smallest possible inference function, but to construct a workable and controlled setting in which the implementation can be tested and the behavior of amortized inference can be examined.

Simulation experiments are carried out in the Jupyter notebook `main-sim-avem-diag-cov.ipynb` performs simulations where EM and AVEM are compared. The results are also discussed in § 4.2 below.

3.2 Relation with auto-encoding variational Bayes

The AVEM procedure discussed above adopts the central amortization idea of auto-encoding variational Bayes, Kingma and Welling (2014) [16]: Local variational parameters are generated by a shared inference network rather than optimized independently for every observation. However, our implementation is not a variational autoencoder in the pure sense. The mixture parameters remain point estimates, the latent allocation variables are discrete, and the ELBO is evaluated directly rather than through the reparameterization-based stochastic gradient estimator employed by Kingma and Welling. These ideas that are a natural continuation of the present work.

4 Simulation experiments

4.1 The classical vs Bayesian approaches

The Jupyter notebook `main-sim-ci.ipynb` contains the simulation experiments described here. The simulation study was initially motivated by the comparison between two conceptually different approaches to inference for a mixture-of-Gaussians model. In the classical approach, the EM algorithm provides maximum-likelihood point estimates and confidence intervals derived from the observed information matrix. In the Bayesian approach, coordinate-ascent variational inference approximates a posterior distribution and consequently provides posterior means, credible intervals, posterior probability statements, and a posterior predictive distribution. Simulating data from a known model makes it possible to compare these procedures under controlled conditions, since the true parameter values and component allocations are available.

An important conclusion of the exercise is that the comparison cannot be reduced to the widths or coverage rates of the reported intervals. A frequentist confidence interval and a Bayesian credible interval have different interpretations: the former is designed to achieve repeated-sampling coverage of a fixed parameter, whereas the latter contains a specified amount of probability under an approximate posterior distribution, conditional on the data and the prior. The results also show that prior specification can materially affect variational point estimates and credible intervals. In particular, an initially conspicuous bias and undercoverage for one precision parameter was substantially

reduced after weakening the prior. Thus, apparently poor variational coverage need not indicate an implementation error; it may reveal the trade-off between the prior and the data-generating parameter values.

A broader comparison should therefore consider several complementary criteria. These include bias in point estimates, classification accuracy, interval width and repeated-sampling coverage, predictive log density on independent test observations, and posterior probability statements such as

$$\Pr_q(\pi_2 > \pi_1 \mid \mathbf{x}) \quad \text{or} \quad \Pr_q(\mu_{2d} > \mu_{1d} \mid \mathbf{x}).$$

Predictive density is particularly useful because it evaluates the fitted distributions on a common out-of-sample scale: EM supplies a plug-in predictive distribution, whereas the Bayesian procedure can integrate predictions over the approximate posterior uncertainty. Posterior probability statements, on the other hand, illustrate a type of uncertainty quantification that EM does not provide directly.

4.2 Amortized inference

As introduced in § 3 above, the AVEM procedure retains point estimates for the mixture parameters and replaces the exact analytical E-step with a neural-network approximation to the responsibilities. The simulations conducted in the Jupyter notebook `main-sim-avem-diag-cov.ipynb` compare the Expectation-Maximization (EM) algorithm and the Amortized Variational EM algorithm. A Mixture-of-Gaussians model is fitted to data generated from the same model. Two initializations of the algorithms are considered: random and k-means.

For the clearly separated components considered, EM gives accurate and stable estimates and is substantially faster. Single-start AVEM occasionally converges to inferior solutions, whereas selecting the largest ELBO among four neural-network initializations produces parameter estimates and classification accuracy very close to those of EM, at a considerably higher computational cost.

The simulation results are encouraging in the intended pedagogical sense. In most of the selected best-of-four AVEM fits, the classification boundary generated by the neural network closely follows the exact quadratic boundary in the region occupied by the observations. The estimated mixture parameters and classification accuracy are also close to those obtained with standard EM. These findings indicate that the implemented inference network is capable of learning a useful approximation to the locally optimal responsibility mapping.

The log-odds plots complement the numerical summaries by examining what the amortized network has learned. In the empirically relevant region containing the observations, the learned mapping usually approximates closely the exact quadratic posterior log-odds implied by the fitted Gaussian mixture. Larger discrepancies arise

mainly away from the data, where the ELBO places little constraint on the network and the flexibility of the ReLU architecture permits unnecessary extrapolation.

The results should be interpreted within the scope of the simulation, where relatively strong separated components are used. Besides, reproducing the quadratic is not so demanding if we consider that the features $(1, \mathbf{x}, \mathbf{x}^2)$ passed to the neural-network are supposed to contain the information required by the exact solution.

5 Application: Weather images clustering

As mentioned before, motivated by the paper by Blei, Kucukelbir, and McAuliffe [8] the Bayesian Mixture-of-Gaussians model was applied to the clustering of an image dataset. The data set is the *Weather Image Recognition Dataset* available on <https://www.kaggle.com/datasets/jehanbhathena/weather-dataset>.

The notebook `main-weather-image-clustering.ipynb` presents an application of variational inference to the unsupervised clustering of the weather-image dataset. The aim is to connect the theoretical study of Bayesian Gaussian mixture models and CAVI with a concrete image-analysis example. Each image is first transformed into a fixed-dimensional colour representation: the red, green, and blue channels are extracted, converted to the $0, \dots, 255$ intensity scale, summarized by 32 normalized histogram bins per channel and concatenated into a 96-dimensional feature vector.

Since the histogram matrix is high-dimensional and contains strong dependencies among neighbouring bins and among the normalized channel histograms, the notebook applies principal component analysis (PCA) before fitting the mixture models. The final analysis uses the first 20 principal components and $K = 10$ mixture components. This choice gives a manageable reduced representation while retaining more colour-intensity information than an earlier choice $D = 10$. Two Bayesian mixture specifications are then fitted: a Normal-Gamma model, which uses a diagonal precision structure, and a Normal-Wishart model, which allows full covariance structure between the principal components. The fitting is performed with a CAVI implementation developed. Convergence is assessed through the evolution of the ELBO.

The results are summarized through posterior component weights, effective cluster sizes, average RGB histograms, and collages of the images assigned to each fitted component. The collages show that the inferred groups are not simply clusters of isolated dominant colours. Instead, the components are better interpreted as broad colour-intensity patterns: bright and pale images, grey or snowy scenes, dark saturated images, among others. This outcome is consistent with the feature representation, because RGB histograms encode brightness, saturation, contrast, and intensity in each channel distributions as much as perceived colour tone or hue.

6 Conclusions and future work

Concluding remarks

This project has studied variational inference through the concrete and progressively richer setting of Gaussian mixture models. Starting from the contrast between exact Bayesian inference in linear Gaussian state-space models and approximate inference in more general models, the work developed the main ideas of variational Bayes as an optimization-based alternative to Monte Carlo methods. The central technical object throughout the project was the evidence lower bound, whose explicit derivation made it possible to connect the mathematical formulation of the model with implementable coordinate-ascent algorithms. The resulting R implementation covers EM, variational EM, CAVI for Bayesian Gaussian mixtures, amortized responsibility mappings, and extensions to autoregressive mixture models.

The simulation experiments highlighted that the comparison between classical and Bayesian procedures cannot be reduced to a single criterion. EM provides maximum-likelihood point estimates and confidence intervals based on the observed information matrix, whereas variational Bayes provides an approximate posterior distribution, credible intervals, posterior probability statements, and posterior predictive summaries. The experiments showed that prior specification can have a visible effect on variational estimates and interval behavior, and that predictive log density, classification accuracy, bias, interval width, and coverage all give complementary information. The amortized variational EM experiments further illustrated that replacing local responsibilities by a shared inference map is feasible, but introduces an additional approximation and optimization problem. In simple separated mixtures, the neural-network responsibility map could approximate the exact quadratic log-odds in the region occupied by the data, at a higher computational cost than standard EM.

The image-clustering application provided a useful empirical counterpart to the mixture-model example in Blei, Kucukelbir, and McAuliffe. The weather images were represented through normalized RGB histograms, reduced by PCA, and clustered with Normal-Gamma and Normal-Wishart variational mixture models. The results were not simply clusters of isolated dominant colours. Instead, the fitted components were better interpreted as broad colour-intensity patterns involving brightness, greyness, saturation and contrast. This outcome was informative because it showed both the usefulness and the limitations of the chosen feature representation: RGB histograms are simple and interpretable, but they discard spatial structure and do not directly encode semantic weather categories. The fact that both models converged without apparent numerical difficulties, even for $D = 20$ principal components and an implementation written largely from scratch in R, is encouraging for the practical viability of the code

developed in the project.

Future work

The most immediate direction for work is to refine the implementation and conduct larger-scale simulation experiments that include a broader grid of parameter values of the data-generating process.

On the methodological side, the ideas studied in this project provide a basis for moving toward the more flexible class of models of variational autoencoders. The EM, CAVI, and amortized-responsibility implementations make explicit the role of latent variables, variational posterior approximations, ELBO optimization, and inference mappings. These are precisely the core elements of **variational autoencoders**.

A further direction for future work would be to investigate alternative amortized inference mappings. In this project, a relatively simple neural network was used to approximate the responsibilities of the mixture model. This was appropriate for a first controlled experiment, especially because in the Gaussian mixture case the ideal posterior log-odds have a known quadratic structure. The same strategy could in principle be adapted to richer inference mappings, for example convolutional neural networks. For example, in the weather-image application, this would allow the inference map to handle local spatial patterns present in the images.

The previous idea is also related to the concept of **structured variational inference** that was found in the literature. Instead of approximating each local latent variable separately and independently, the variational approximation is designed so that it captures known structure in the data. For example, in image applications the variational family or inference network could be built to exploit spatial structure, while in time-series applications it could be designed to reflect temporal dependence.

The reference Girin, Leglaive, Bie and Diard, Hueber and Alameda-Pineda (2021) [17] appears also relevant for future work. It provides a review of **dynamic variational autoencoders** and may offer a framework for studying both the linear autoregressive mixture models discussed here and a class of more flexible nonlinear dynamic models with latent variables.

References

- [1] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [2] Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.
- [3] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Oxford Statistical Science Series. Oxford University Press, 2001.
- [4] S. Särkkä and L. Svensson. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2 edition, 2023.
- [5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025.
- [6] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, January 2006.
- [7] Charles C. Margossian and David M. Blei. Amortized variational inference: When and why? In Negar Kiyavash and Joris M. Mooij, editors, *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pages 2434–2449. PMLR, 15–19 Jul 2024.
- [8] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- [9] S.J. Roberts and W.D. Penny. Variational Bayes for generalized autoregressive models. *IEEE Transactions on Signal Processing*, 50(9):2245–2257, 2002.
- [10] Harri Lappalainen and James W. Miskin. Ensemble learning. In Mark Girolami, editor, *Advances in Independent Component Analysis*, Perspectives in Neural Computing, pages 75–92. Springer, London, 2000.
- [11] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999.
- [12] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

-
- [13] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, May 2013.
 - [14] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, September 1951.
 - [15] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.
 - [16] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
 - [17] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends in Machine Learning*, 15(1–2):1–175, 2021.

Appendix

Note: This appendix adopts a deliberately concise and schematic style. It is an organized compilation of the principal steps and expressions studied in the project. For clarity and visibility, mathematical expressions are generally displayed on separate lines rather than included inline surrounded by text. In L^AT_EX terminology: display math is preferred to inline.

The expressions collected in this appendix were developed at different stages of the project and were based on different sources in the literature. Consequently, the notation is not fully uniform across all sections. For example, responsibilities are in some models denoted τ and in other γ or r . Standard notation is used anyway and some notational changes may be subtle or go unnoticed. They are therefore not expected to affect the exposition.

A EM algorithm. Full covariance matrix model.

A.1 The Model.

D denotes the dimension of each observation $\mathbf{x}_n$, K is the number of mixture components, and N is the sample size. The latent allocation vector $\mathbf{z}_n$ is a 1-of- K binary vector, so that $z_{nk} \in \{0, 1\}$ and $\sum_{k=1}^K z_{nk} = 1$.

The mixing probabilities are denoted $\boldsymbol{\pi}$.

A full-covariance Gaussian mixture is considered. The component density is:

$$p(\mathbf{x}_n \mid z_{nk} = 1) = \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where

$$\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1}.$$

An alternative model (computationally simpler) is the diagonal-covariance Gaussian mixture: Conditional on the mixture component, the coordinates of $\mathbf{x}_n = (x_{n1}, \dots, x_{nD})$ are conditionally independent Gaussian variables. In this case $\boldsymbol{\Lambda}_k$ are diagonal.

For component k , $\boldsymbol{\mu}_k = (\mu_{k1}, \dots, \mu_{kD})$ and $\boldsymbol{\beta}_k = (\beta_{k1}, \dots, \beta_{kD})$ will denote respectively the mean vector and the precision vectors.

Let denote the vector of parameters as $\boldsymbol{\theta} = \boldsymbol{\theta}_1 \cup \boldsymbol{\pi}$, where $\boldsymbol{\theta}_1 = \{\boldsymbol{\mu}, \boldsymbol{\beta}\}$.

It is convenient to define a latent indicator vector $\mathbf{z}_n = (z_{n1}, \dots, z_{nK})$ for each observation $\mathbf{x}_n$. Conditional on the mixing probabilities $\boldsymbol{\pi}$, the allocation variable $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_N)$ satisfies:

$$p(\mathbf{Z} \mid \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}}.$$

Conditional on $\mathbf{Z}$, the component means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ and the component precision matrices $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_k\}_{k=1}^K$, the observed data have density:

$$p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}.$$

The complete joint density factorizes as:

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\mathbf{Z} \mid \boldsymbol{\pi}) p(\boldsymbol{\pi}) p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) p(\boldsymbol{\Lambda}).$$

A.2 Complete-data likelihood.

The joint likelihood of the observed data and the latent allocations, conditional on the parameters, is given by:

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) &= p(\mathbf{Z} \mid \boldsymbol{\pi}) p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \\ &= \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}. \end{aligned}$$

The complete-data log-likelihood is therefore:

$$\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left[\log \pi_k + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right].$$

A.3 EM objective function.

Since $\mathbf{Z}$ is not observed, the following **expected** log-likelihood is maximized:

$$\begin{aligned} Q(\theta \mid \theta^{old}) &:= Q(\theta \mid \mathbf{x}, \theta^{(k-1)}) = \mathbb{E}_{\mathbf{Z} \mid \mathbf{x}, \theta^{old}} \log p(\mathbf{x}, \mathbf{Z} \mid \theta) \\ &= \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \mathbf{x}, \theta^{(k-1)}) \log p(\mathbf{x}, \mathbf{Z} \mid \theta) \end{aligned}$$

This expectation is computed conditional on *old* parameters $\theta^{(k-1)} = \theta_1^{(k-1)} \cup \boldsymbol{\pi}^{(k-1)}$ from a previous iteration (starting at some initial $\theta_1^{(0)}, \boldsymbol{\pi}^{(0)}$).

Let's denote:

$$\tau_{nk} = \mathbb{E}_{\theta^{old}}[z_{nk} \mid \mathbf{x}],$$

so that, the complete-data expected log-likelihood Q -function is:

$$Q(\theta \mid \theta^{old}) = \sum_{n=1}^N \sum_{k=1}^K \tau_{nk} \left[\log \pi_k + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right].$$

A.4 Expectation step.

Applying Bayes' rule, the posterior probability of component k at time n is:

$$\begin{aligned} \tau_{nk} &= p(Z_n = k \mid x_n, \theta^{old}) \\ &= \frac{\pi_k^{old} (2\pi)^{-D/2} \det(\boldsymbol{\Lambda}_k^{old})^{1/2} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k^{old})^\top \boldsymbol{\Lambda}_k^{old} (\mathbf{x} - \boldsymbol{\mu}_k^{old})\right)}{\sum_{r=1}^m \pi_r^{old}(\dots)}. \end{aligned}$$

In the implementation, we consider just the logs of the numerator and normalize using softmax.

These are the *responsibilities*. They are analogous to the CAVI responsibilities γ_{nk} , but they are computed using parameter estimates rather than variational expectations.

A.5 Maximization step.

Equation to zero the derivatives of the expected log-likelihood Q and solving for the parameters of the model, we get the update equations. They are run iteratively until convergence.

Update for $\boldsymbol{\pi}$.

The part of Q involving π_k is:

$$\sum_{n=1}^N \sum_{k=1}^K \tau_{nk} \log \pi_k, \quad \text{subject to} \quad \sum_{k=1}^K \pi_k = 1.$$

Using a Lagrange multiplier gives:

$$\pi_k^{new} = \frac{N_k}{N}, \quad \text{where} \quad N_k = \sum_{n=1}^N \tau_{nk}.$$

Update for μ_k .

The relevant part of Q is:

$$-\frac{1}{2} \sum_{n=1}^N \tau_{nk} (\mathbf{x}_n - \mu_k)^\top \Lambda_k (\mathbf{x}_n - \mu_k).$$

Differentiating gives:

$$\frac{\partial Q}{\partial \mu_k} = \sum_{n=1}^N \tau_{nk} \Lambda_k (\mathbf{x}_n - \mu_k) = 0.$$

Solving yields:

$$\mu_k^{new} = \frac{\sum_{n=1}^N \tau_{nk} \mathbf{x}_n}{\sum_{n=1}^N \tau_{nk}}$$

Update for Λ_k .

The part of Q involving β_s is:

$$\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K \tau_{nk} \log |\Lambda_k| - \frac{1}{2} (\mathbf{x}_n - \mu_k)^\top \Lambda_k (\mathbf{x}_n - \mu_k)$$

Differentiating gives:

$$\frac{\partial Q}{\partial \Lambda_k} = \frac{1}{2} \left(N_k \Lambda_k^{-1} - \sum_{n=1}^N \tau_{nk} (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^\top \right), \quad \text{where} \quad N_k = \sum_{n=1}^N \tau_{nk}.$$

Solving gives:

$$\Lambda_k^{-1, new} = \frac{1}{N_k} \sum_{n=1}^N \tau_{nk} (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^\top.$$

B Amortized Variational EM algorithm

B.1 Variational distribution for the allocations

The EM algorithm can be expressed in variational terms by introducing a distribution $q(\mathbf{Z})$ over the latent allocation variables. For independently distributed observations, consider the categorical factorization

$$q(\mathbf{Z}) = \prod_{n=1}^N q_n(\mathbf{z}_n) = \prod_{n=1}^N \prod_{k=1}^K r_{nk}^{z_{nk}},$$

where

$$r_{nk} \geq 0, \quad \sum_{k=1}^K r_{nk} = 1.$$

The quantities r_{nk} are variational responsibilities. In contrast with a fully Bayesian variational treatment, the mixture parameters

$$\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\}$$

are retained as point estimates. The only distribution approximated variationally is the posterior distribution of the latent allocations.

B.2 Evidence lower bound

For any distribution $q(\mathbf{Z})$, the observed-data log-likelihood admits the decomposition

$$\log p(\mathbf{X} \mid \boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})),$$

where

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})] - \mathbb{E}_q[\log q(\mathbf{Z})].$$

Since the Kullback–Leibler divergence is nonnegative,

$$\mathcal{L}(q, \boldsymbol{\theta}) \leq \log p(\mathbf{X} \mid \boldsymbol{\theta}).$$

For the Gaussian mixture model, the ELBO is

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left[\log \pi_k + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) - \log r_{nk} \right],$$

with the convention $0 \log 0 = 0$. The first terms constitute the expected complete-data log-likelihood, while

$$- \sum_{n=1}^N \sum_{k=1}^K r_{nk} \log r_{nk}$$

is the entropy of the variational allocation distribution.

It is convenient to define the component log-scores

$$s_{nk}(\boldsymbol{\theta}) = \log \pi_k + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k).$$

The ELBO can then be written compactly as

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{n=1}^N \sum_{k=1}^K r_{nk} [s_{nk}(\boldsymbol{\theta}) - \log r_{nk}].$$

B.3 Variational E-step

The variational E-step holds the current model parameters $\boldsymbol{\theta}^{(t)}$ fixed and solves

$$q^{(t+1)} = \arg \max_{q \in \mathcal{Q}} \mathcal{L}(q, \boldsymbol{\theta}^{(t)}),$$

where $\mathcal{Q}$ denotes the chosen variational family.

If all categorical probabilities r_{nk} can vary independently, maximizing the ELBO subject to their simplex constraints gives

$$r_{nk}^{(t+1)} = \frac{\exp \{s_{nk}(\boldsymbol{\theta}^{(t)})\}}{\sum_{j=1}^K \exp \{s_{nj}(\boldsymbol{\theta}^{(t)})\}}.$$

These probabilities are exactly

$$r_{nk}^{(t+1)} = p(z_{nk} = 1 \mid \mathbf{x}_n, \boldsymbol{\theta}^{(t)}).$$

Consequently, for an unrestricted categorical variational family, the variational E-step is identical to the ordinary EM E-step. At its optimum,

$$q^{(t+1)}(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)}),$$

so that

$$\text{KL}(q^{(t+1)}(\mathbf{Z}) \parallel p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{(t)})) = 0$$

and hence

$$\mathcal{L}(q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = \log p(\mathbf{X} \mid \boldsymbol{\theta}^{(t)}).$$

Thus, variational EM differs substantially from standard EM only when $\mathcal{Q}$ restricts the responsibilities or otherwise prevents q from representing the exact posterior.

B.4 Variational M-step

Given the updated variational responsibilities, the M-step solves

$$\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{L}(q^{(t+1)}, \boldsymbol{\theta}).$$

The entropy of $q^{(t+1)}$ does not depend on $\boldsymbol{\theta}$. Therefore, the M-step maximizes the same weighted complete-data log-likelihood as standard EM, replacing the exact responsibilities τ_{nk} by the variational responsibilities r_{nk} .

Define

$$N_k = \sum_{n=1}^N r_{nk}.$$

The mixing-probability update is

$$\pi_k^{\text{new}} = \frac{N_k}{N}.$$

The component-mean update is

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} \mathbf{x}_n.$$

For the full-covariance model, the precision update is determined by

$$(\boldsymbol{\Lambda}_k^{\text{new}})^{-1} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^\top.$$

For the diagonal-covariance model, this becomes, coordinate by coordinate,

$$(\lambda_{kd}^{\text{new}})^{-1} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (x_{nd} - \mu_{kd}^{\text{new}})^2.$$

The variational E-step and the analytical M-step are alternated until the relative change in the ELBO is sufficiently small. If each step attains a value no lower than that of the preceding iteration, the ELBO is nondecreasing.

C Amortized variational inference for the responsibilities

C.1 Amortization of the local variational parameters

In an unrestricted variational E-step, every observation has its own categorical parameter vector

$$\mathbf{r}_n = (r_{n1}, \dots, r_{nK}).$$

Amortization replaces these N separately optimized vectors by a common inference mapping. Let $\boldsymbol{\phi}(\mathbf{x}_n)$ denote a vector of features extracted from observation $\mathbf{x}_n$. The responsibilities are parameterized as

$$\mathbf{r}_n(\boldsymbol{\eta}) = q_{\boldsymbol{\eta}}(\mathbf{z}_n \mid \boldsymbol{\phi}(\mathbf{x}_n)),$$

where $\boldsymbol{\eta}$ is shared by all observations. The resulting variational family is

$$q_{\boldsymbol{\eta}}(\mathbf{Z} \mid \mathbf{X}) = \prod_{n=1}^N \prod_{k=1}^K r_{nk}(\boldsymbol{\eta})^{z_{nk}}.$$

What is amortized is the computation of the local categorical parameters: instead of optimizing N unrelated responsibility vectors, a single fitted function maps the features of each observation to its responsibilities.

In the AAIMF implementation, the inference mapping is estimated separately for each data set. Amortization therefore takes place primarily across observations within a fitted sample, rather than across many independently observed data sets as in some applications of amortized variational inference.

C.2 Softmax parameterization

Let

$$\mathbf{u}_\eta(\mathbf{x}_n) = (u_{n1}, \dots, u_{nK})$$

denote the logits produced by the inference mapping. The responsibilities are

$$r_{nk}(\boldsymbol{\eta}) = \frac{\exp(u_{nk})}{\sum_{j=1}^K \exp(u_{nj})}.$$

Since softmax is invariant to adding the same quantity to every logit, the first component can be used as the baseline:

$$u_{n1} = 0.$$

Only $K - 1$ nonbaseline logits then need to be parameterized.

For a linear amortization map,

$$u_{n,k+1} = \boldsymbol{\phi}(\mathbf{x}_n)^\top \boldsymbol{\eta}_k, \quad k = 1, \dots, K - 1.$$

Collecting the feature vectors into a matrix Φ , the nonbaseline logits are obtained from

$$\Phi \boldsymbol{\eta},$$

where the columns of $\boldsymbol{\eta}$ contain the coefficient vectors for the $K - 1$ nonbaseline components.

C.3 Neural-network inference mapping

A more flexible alternative uses a separate feed-forward neural network for each nonbaseline logit. For $k = 1, \dots, K - 1$, the network employed in the implementation has two hidden ReLU layers:

$$\mathbf{h}_{nk}^{(1)} = \text{ReLU} \left(\boldsymbol{\phi}(\mathbf{x}_n)^\top \mathbf{W}_{1k} + \mathbf{b}_{1k} \right),$$

$$\mathbf{h}_{nk}^{(2)} = \text{ReLU} \left(\mathbf{h}_{nk}^{(1)} \mathbf{W}_{2k} + \mathbf{b}_{2k} \right),$$

$$u_{n,k+1} = \mathbf{h}_{nk}^{(2)} \mathbf{W}_{3k} + b_{3k}.$$

The complete set of inference-network parameters is

$$\boldsymbol{\eta} = \{\mathbf{W}_{1k}, \mathbf{b}_{1k}, \mathbf{W}_{2k}, \mathbf{b}_{2k}, \mathbf{W}_{3k}, b_{3k}\}_{k=1}^{K-1}.$$

The ReLU activation is

$$\text{ReLU}(v) = \max(0, v).$$

The network is therefore nonlinear in the input features and can construct piecewise-defined inference mappings, while its softmax output guarantees nonnegative responsibilities that sum to one.

C.4 Amortized variational E-step

Under amortization, the variational E-step becomes an optimization over the shared parameters $\boldsymbol{\eta}$:

$$\boldsymbol{\eta}^{(t+1)} = \arg \max_{\boldsymbol{\eta}} \mathcal{L} \left(q_{\boldsymbol{\eta}}, \boldsymbol{\theta}^{(t)} \right).$$

More explicitly,

$$\boldsymbol{\eta}^{(t+1)} = \arg \max_{\boldsymbol{\eta}} \sum_{n=1}^N \sum_{k=1}^K r_{nk}(\boldsymbol{\eta}) \left[s_{nk}(\boldsymbol{\theta}^{(t)}) - \log r_{nk}(\boldsymbol{\eta}) \right].$$

Unlike the exact E-step, this optimization generally has no closed-form solution. In the implementation, the network parameters are flattened into a vector and optimized numerically, for example using BFGS. During this optimization, the mixture parameters $\boldsymbol{\theta}^{(t)}$ remain fixed. The analytical M-step is then applied using the responsibilities generated by the optimized network:

$$\mathbf{r}_n^{(t+1)} = \mathbf{r}_n \left(\boldsymbol{\eta}^{(t+1)} \right),$$

$$\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{L} \left(q_{\boldsymbol{\eta}^{(t+1)}}, \boldsymbol{\theta} \right).$$

This gives the coordinate-wise scheme

$$\boldsymbol{\eta}^{(t)} \longrightarrow \mathbf{R}^{(t+1)} \longrightarrow \boldsymbol{\theta}^{(t+1)},$$

where $\mathbf{R}$ is the matrix of amortized responsibilities.

C.5 Initialization

The inference-network parameters may be initialized randomly. Alternatively, k-means can provide preliminary target responsibilities. If c_n denotes the cluster assigned to observation n , smoothed targets can be defined as

$$\tilde{r}_{nk} = \begin{cases} \rho, & k = c_n, \\ \frac{1 - \rho}{K - 1}, & k \neq c_n, \end{cases}$$

where

$$\frac{1}{K} \leq \rho \leq 1.$$

The network is pretrained by minimizing the cross-entropy between $\tilde{\mathbf{r}}_n$ and $\mathbf{r}_n(\boldsymbol{\eta})$, optionally including a weight penalty:

$$-\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \tilde{r}_{nk} \log r_{nk}(\boldsymbol{\eta}) + \frac{\lambda}{2} \|\boldsymbol{\eta}\|^2.$$

Because neural-network optimization remains nonconvex, k-means pretraining does not make the initialization fully deterministic. Several network starts may be fitted, retaining the solution with the highest final ELBO.

D CAVI algorithm. Normal-Wishart model.

The following derivation of the Coordinate-Ascent Variational Inference algorithm follows Blei, Kucukelbir and McAuliffe (2017) [8] and Bishop (2006) [6, § 10.2].

D.1 Description of the model

Below, D denotes the dimension of each observation $\mathbf{x}_n$, K is the number of mixture components, and N is the sample size. The latent allocation vector $\mathbf{z}_n$ is a 1-of- K binary vector, so that $z_{nk} \in \{0, 1\}$ and $\sum_{k=1}^K z_{nk} = 1$.

The Bayesian Gaussian mixture model introduces a latent indicator vector $\mathbf{z}_n = (z_{n1}, \dots, z_{nK})$ for each observation $\mathbf{x}_n$. Conditional on the mixing probabilities $\boldsymbol{\pi}$, the allocation variables satisfy

$$p(\mathbf{Z} \mid \boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}}.$$

Conditional on $\mathbf{Z}$, the component means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$, and the component precision matrices $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_k\}_{k=1}^K$, the observed data have density

$$p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}.$$

The complete joint density factorizes as:

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\mathbf{Z} \mid \boldsymbol{\pi}) p(\boldsymbol{\pi}) p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda}) p(\boldsymbol{\Lambda}).$$

D.2 Prior densities

- **Mixing probabilities $\boldsymbol{\pi}$.** [Dirichlet](#) distribution:

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0) \prod_{k=1}^K \pi_k^{\alpha_{0k}-1}, \quad \text{where} \quad C(\boldsymbol{\alpha}_0) = \frac{\Gamma(\hat{\alpha}_0)}{\prod_{k=1}^K \Gamma(\alpha_{0k})}, \quad \hat{\alpha}_0 = \sum_{k=1}^K \alpha_{0k}.$$

In the symmetric case $\alpha_{0k} = \alpha_0$:

$$p(\boldsymbol{\pi}) = \frac{\Gamma(K\alpha_0)}{\Gamma(\alpha_0)^K} \prod_{k=1}^K \pi_k^{\alpha_0-1}.$$

- **Mixture components $(\boldsymbol{\mu}, \boldsymbol{\Lambda})$.** [Gaussian-Wishart](#) distribution.

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{k=1}^K p(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) p(\boldsymbol{\Lambda}_k) :$$

- Gaussian conditional prior for the component means:

$$\begin{aligned} p(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) &= \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_0, (\beta_0 \boldsymbol{\Lambda}_k)^{-1}) \\ &= \left(\frac{\beta_0}{2\pi} \right)^{D/2} |\boldsymbol{\Lambda}_k|^{1/2} \exp \left\{ -\frac{\beta_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \right\}, \end{aligned}$$

– Wishart prior for the precision matrix:

$$\begin{aligned} p(\mathbf{\Lambda}_k) &= \mathcal{W}(\mathbf{\Lambda}_k \mid \mathbf{W}_0, \nu_0) \\ &= B(\mathbf{W}_0, \nu_0) |\mathbf{\Lambda}_k|^{(\nu-D-1)/2} \exp \left\{ -\frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \mathbf{\Lambda}_k) \right\}, \\ \text{where } B(\mathbf{W}_0, \nu_0) &= |\mathbf{W}|^{-\nu_0/2} \left[2^{\nu_0 D/2} \pi^{D(D-1)/4} \prod_{i=1}^D \Gamma \left(\frac{\nu_0 + 1 - i}{2} \right) \right]^{-1}. \end{aligned}$$

D.3 Expression of the log-joint density

The complete log-joint density is

$$\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}) = \log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \mathbf{\Lambda}) + \log p(\mathbf{Z} \mid \boldsymbol{\pi}) + \log p(\boldsymbol{\pi}) + \log p(\boldsymbol{\mu} \mid \mathbf{\Lambda}) + \log p(\mathbf{\Lambda}).$$

The likelihood contribution is

$$\log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \mathbf{\Lambda}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left\{ \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \mathbf{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right\}.$$

The latent allocation contribution is

$$\log p(\mathbf{Z} \mid \boldsymbol{\pi}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \log \pi_k.$$

The Dirichlet prior contribution is

$$\log p(\boldsymbol{\pi}) = \log C(\boldsymbol{\alpha}_0) + \sum_{k=1}^K (\alpha_{0k} - 1) \log \pi_k.$$

The Gaussian prior contribution is

$$\log p(\boldsymbol{\mu} \mid \mathbf{\Lambda}) = \sum_{k=1}^K \left\{ \frac{D}{2} \log \beta_0 - \frac{D}{2} \log(2\pi) + \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{\beta_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \mathbf{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \right\}.$$

The Wishart prior contribution is

$$\log p(\mathbf{\Lambda}) = \sum_{k=1}^K \left\{ \log B(\mathbf{W}_0, \nu_0) + \frac{\nu_0 - D - 1}{2} \log |\mathbf{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \mathbf{\Lambda}_k) \right\}.$$

D.4 Expression of the joint log-likelihood

The joint likelihood of the observed data and the latent allocations, conditional on the parameters, is

$$p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}) = p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \mathbf{\Lambda}) p(\mathbf{Z} \mid \boldsymbol{\pi}).$$

Therefore,

$$\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Lambda}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left\{ \log \pi_k + \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \mathbf{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right\}.$$

D.5 Variational family

Below, the subscripts are omitted to simplify notation. $q_{\mathbf{Z}}(\mathbf{Z})q_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda})$; i.e. the variational density q may differ for each factor. The mean-field variational distribution factorizes as:

$$q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z})q(\boldsymbol{\pi})q(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})q(\boldsymbol{\Lambda}).$$

The variational posteriors are:

$$\begin{aligned} q(\mathbf{Z}) &= \prod_{n=1}^N \prod_{k=1}^K r_{nk}^{z_{nk}}, \quad \text{with } r_{nk} \geq 0, \quad \sum_{k=1}^K r_{nk} = 1, \\ q(\boldsymbol{\pi}) &= \text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}), \\ q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \prod_{k=1}^K q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k), \\ q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) &= q(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k)q(\boldsymbol{\Lambda}_k), \\ q(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) &= \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}), \\ q(\boldsymbol{\Lambda}_k) &= \mathcal{W}(\boldsymbol{\Lambda}_k \mid \mathbf{W}_k, \nu_k). \end{aligned}$$

D.6 Expected log-density under q , including constants

Observe that we have the following expectations:

$$\begin{aligned} \mathbb{E}_q[\log \pi_k] &= \psi(\alpha_k) - \psi(\hat{\alpha}), \quad \hat{\alpha} = \sum_{j=1}^K \alpha_j, \\ \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] &= \sum_{i=1}^D \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D \log 2 + \log |\mathbf{W}_k|, \\ \mathbb{E}_q[\boldsymbol{\Lambda}_k] &= \nu_k \mathbf{W}_k, \\ \mathbb{E}_q\left[(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k)\right] &= \frac{D}{\beta_k} + \nu_k (\mathbf{x}_n - \mathbf{m}_k)^\top \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k). \end{aligned}$$

We have:

$$\begin{aligned}
 \mathbb{E}_q[\log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] &= \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left\{ \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk} \right\}, \\
 \text{with } Q_{nk} &:= \frac{D}{\beta_k} + \nu_k (\mathbf{x}_n - \mathbf{m}_k)^\top \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k), \\
 \mathbb{E}_q[\log p(\mathbf{Z} \mid \boldsymbol{\pi})] &= \sum_{n=1}^N \sum_{k=1}^K r_{nk} \mathbb{E}_q[\log \pi_k], \\
 \mathbb{E}_q[\log p(\boldsymbol{\pi})] &= \log C(\boldsymbol{\alpha}_0) + \sum_{k=1}^K (\alpha_{0k} - 1) \mathbb{E}_q[\log \pi_k], \\
 \mathbb{E}_q[\log p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})] &= \sum_{k=1}^K \left\{ \frac{D}{2} \log \beta_0 - \frac{D}{2} \log(2\pi) + \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{\beta_0}{2} M_k \right\}, \\
 \text{with } M_k &:= \frac{D}{\beta_k} + \nu_k (\mathbf{m}_k - \mathbf{m}_0)^\top \mathbf{W}_k (\mathbf{m}_k - \mathbf{m}_0), \\
 \mathbb{E}_q[\log p(\boldsymbol{\Lambda})] &= \sum_{k=1}^K \left\{ \log B(\mathbf{W}_0, \nu_0) + \frac{\nu_0 - D - 1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{1}{2} \nu_k \text{tr}(\mathbf{W}_0^{-1} \mathbf{W}_k) \right\}.
 \end{aligned}$$

D.7 Entropy terms, including constants

The entropy of the categorical variational distribution $q(\mathbf{Z})$ is

$$\mathcal{H}[q(\mathbf{Z})] = -\mathbb{E}_q[\log q(\mathbf{Z})] = -\sum_{n=1}^N \sum_{k=1}^K r_{nk} \log r_{nk}.$$

The entropy of the Dirichlet variational distribution $q(\boldsymbol{\pi})$ is

$$\mathcal{H}[q(\boldsymbol{\pi})] = \log B_{\text{Dir}}(\boldsymbol{\alpha}) + (\hat{\alpha} - K) \psi(\hat{\alpha}) - \sum_{k=1}^K (\alpha_k - 1) \psi(\alpha_k),$$

where

$$\log B_{\text{Dir}}(\boldsymbol{\alpha}) = \sum_{k=1}^K \log \Gamma(\alpha_k) - \log \Gamma(\hat{\alpha}).$$

The entropy of the Gaussian-Wishart factor can be decomposed as

$$\mathcal{H}[q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)] = \mathbb{E}_{q(\boldsymbol{\Lambda}_k)} [\mathcal{H}[q(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k)]] + \mathcal{H}[q(\boldsymbol{\Lambda}_k)].$$

The conditional Gaussian entropy is

$$\mathbb{E}_{q(\boldsymbol{\Lambda}_k)} [\mathcal{H}[q(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k)]] = \frac{D}{2} \{1 + \log(2\pi)\} - \frac{D}{2} \log \beta_k - \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|].$$

The Wishart entropy is

$$\mathcal{H}[q(\boldsymbol{\Lambda}_k)] = -\log B(\mathbf{W}_k, \nu_k) - \frac{\nu_k - D - 1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] + \frac{\nu_k D}{2}.$$

Hence

$$\begin{aligned}
 \mathcal{H}[q(\boldsymbol{\mu}, \boldsymbol{\Lambda})] &= \sum_{k=1}^K \left[\frac{D}{2} \{1 + \log(2\pi)\} - \frac{D}{2} \log \beta_k - \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] \right. \\
 &\quad \left. - \log B(\mathbf{W}_k, \nu_k) - \frac{\nu_k - D - 1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] + \frac{\nu_k D}{2} \right].
 \end{aligned}$$

Expression of the ELBO, including constants

The evidence lower bound is

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] - \mathbb{E}_q[\log q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})].$$

Equivalently,

$$\begin{aligned} \mathcal{L}(q) = & \mathbb{E}_q[\log p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \mathbb{E}_q[\log p(\mathbf{Z} \mid \boldsymbol{\pi})] + \mathbb{E}_q[\log p(\boldsymbol{\pi})] \\ & + \mathbb{E}_q[\log p(\boldsymbol{\mu} \mid \boldsymbol{\Lambda})] + \mathbb{E}_q[\log p(\boldsymbol{\Lambda})] + \mathcal{H}[q(\mathbf{Z})] + \mathcal{H}[q(\boldsymbol{\pi})] + \mathcal{H}[q(\boldsymbol{\mu}, \boldsymbol{\Lambda})]. \end{aligned}$$

Substituting the terms above gives

$$\begin{aligned} \mathcal{L}(q) = & \sum_{n=1}^N \sum_{k=1}^K r_{nk} \left\{ \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk} \right\} \\ & + \sum_{n=1}^N \sum_{k=1}^K r_{nk} \mathbb{E}_q[\log \pi_k] \\ & + \log C(\boldsymbol{\alpha}_0) + \sum_{k=1}^K (\alpha_{0k} - 1) \mathbb{E}_q[\log \pi_k] \\ & + \sum_{k=1}^K \left\{ \frac{D}{2} \log \beta_0 - \frac{D}{2} \log(2\pi) + \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{\beta_0}{2} M_k \right\} \\ & + \sum_{k=1}^K \left\{ \log B(\mathbf{W}_0, \nu_0) + \frac{\nu_0 - D - 1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{1}{2} \nu_k \text{tr}(\mathbf{W}_0^{-1} \mathbf{W}_k) \right\} \\ & - \sum_{n=1}^N \sum_{k=1}^K r_{nk} \log r_{nk} \\ & + \log B_{\text{Dir}}(\boldsymbol{\alpha}) + (\hat{\alpha} - K) \psi(\hat{\alpha}) - \sum_{k=1}^K (\alpha_k - 1) \psi(\alpha_k) \\ & + \sum_{k=1}^K \left\{ \frac{D}{2} \{1 + \log(2\pi)\} - \frac{D}{2} \log \beta_k - \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \log B(\mathbf{W}_k, \nu_k) \right. \\ & \left. - \frac{\nu_k - D - 1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] + \frac{\nu_k D}{2} \right\}. \end{aligned}$$

This expression includes the normalizing constants from the Gaussian, Dirichlet, and Wishart densities. It can be used directly as a numerical ELBO.

Coordinate-ascent updates obtained by direct differentiation of the ELBO

We now derive the coordinate-ascent variational updates by differentiating the ELBO with respect to each variational block while holding the remaining blocks fixed.

Recall the variational family

$$q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z})q(\boldsymbol{\pi}) \prod_{k=1}^K q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k),$$

where

$$q(\mathbf{Z}) = \prod_{n=1}^N \prod_{k=1}^K r_{nk}^{z_{nk}},$$

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}),$$

$$q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k \mid \mathbf{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k \mid \mathbf{W}_k, \nu_k).$$

D.8 Coordinate-ascent algorithm

Define the effective number of observations allocated to component k by

$$N_k = \sum_{n=1}^N r_{nk}.$$

Define the empirical responsibility-weighted mean by

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} \mathbf{x}_n,$$

whenever $N_k > 0$. Define also the responsibility-weighted scatter matrix

$$\mathbf{S}_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^\top.$$

We use the identities

$$\sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top = N_k \mathbf{S}_k + N_k (\bar{\mathbf{x}}_k - \boldsymbol{\mu}_k)(\bar{\mathbf{x}}_k - \boldsymbol{\mu}_k)^\top.$$

Update for $q(\mathbf{Z})$

Collect from the ELBO only the terms that depend on r_{nk} . For a fixed n , the relevant part is

$$\begin{aligned} \mathcal{L}_n(r_{n1}, \dots, r_{nK}) = & \sum_{k=1}^K r_{nk} \left\{ \mathbb{E}_q[\log \pi_k] + \frac{1}{2} \mathbb{E}_q[\log |\boldsymbol{\Lambda}_k|] - \frac{D}{2} \log(2\pi) \right. \\ & \left. - \frac{1}{2} \mathbb{E}_q \left[(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right] \right\} - \sum_{k=1}^K r_{nk} \log r_{nk}. \end{aligned}$$

The constraint is

$$\sum_{k=1}^K r_{nk} = 1.$$

Introduce a Lagrange multiplier λ_n and define

$$\mathcal{J}_n = \mathcal{L}_n + \lambda_n \left(\sum_{k=1}^K r_{nk} - 1 \right).$$

Let

$$A_{nk} = \mathbb{E}_q[\log \pi_k] + \frac{1}{2} \mathbb{E}_q[\log |\mathbf{\Lambda}_k|] - \frac{D}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_q \left[(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \mathbf{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right].$$

Then

$$\mathcal{J}_n = \sum_{k=1}^K r_{nk} A_{nk} - \sum_{k=1}^K r_{nk} \log r_{nk} + \lambda_n \left(\sum_{k=1}^K r_{nk} - 1 \right).$$

Differentiating with respect to r_{nk} gives

$$\frac{\partial \mathcal{J}_n}{\partial r_{nk}} = A_{nk} - (\log r_{nk} + 1) + \lambda_n.$$

Setting this derivative equal to zero gives

$$\log r_{nk} = A_{nk} + \lambda_n - 1.$$

Hence

$$r_{nk} \propto \exp(A_{nk}),$$

and normalization over k gives

$$r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^K \rho_{nj}},$$

where

$$\rho_{nk} = \exp \left\{ \mathbb{E}_q[\log \pi_k] + \frac{1}{2} \mathbb{E}_q[\log |\mathbf{\Lambda}_k|] - \frac{D}{2} \log(2\pi) - \frac{1}{2} \left[\frac{D}{\beta_k} + \nu_k (\mathbf{x}_n - \mathbf{m}_k)^\top \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k) \right] \right\}.$$

Since the factor $\exp\{-D \log(2\pi)/2\}$ is common to all components for fixed n , it can be omitted in numerical implementations without changing the normalized responsibilities.

The required expectations are

$$\mathbb{E}_q \log \pi_k = \psi(\alpha_k) - \psi(\hat{\alpha}), \quad \hat{\alpha} = \sum_{j=1}^K \alpha_j,$$

$$\mathbb{E}_q \log |\mathbf{\Lambda}_k| = \sum_{i=1}^D \psi \left(\frac{\nu_k + 1 - i}{2} \right) + D \log 2 + \log |\mathbf{W}_k|.$$

Update for $q(\boldsymbol{\pi})$

The terms of the ELBO that depend on $q(\boldsymbol{\pi})$ are

$$\mathcal{L}_\pi = \mathbb{E}_q \log p(\mathbf{Z} \mid \boldsymbol{\pi}) + \mathbb{E}_q \log p(\boldsymbol{\pi}) + \mathcal{H}[q(\boldsymbol{\pi})].$$

Combining terms gives

$$\mathcal{L}_\pi = \sum_{k=1}^K (N_k + \alpha_{0k} - 1) \mathbb{E}_q \log \pi_k + \mathcal{H}[q(\boldsymbol{\pi})] + \text{const.}$$

Since

$$q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} \mid \boldsymbol{\alpha}),$$

its entropy can be written as

$$\mathcal{H}[q(\boldsymbol{\pi})] = -\mathbb{E}_q \log q(\boldsymbol{\pi}).$$

Therefore,

$$\mathcal{L}_\pi = \mathbb{E}_q \left[\sum_{k=1}^K (N_k + \alpha_{0k} - 1) \log \pi_k - \sum_{k=1}^K (\alpha_k - 1) \log \pi_k \right] - \log C(\boldsymbol{\alpha}) + \text{const.}$$

Define

$$\alpha_k^* = \alpha_{0k} + N_k.$$

Then

$$\mathcal{L}_\pi = \sum_{k=1}^K (\alpha_k^* - \alpha_k) \mathbb{E}_q [\log \pi_k] - \log C(\boldsymbol{\alpha}) + \text{const.}$$

Differentiating with respect to α_j gives

$$\frac{\partial \mathcal{L}_\pi}{\partial \alpha_j} = \sum_{k=1}^K (\alpha_k^* - \alpha_k) \frac{\partial}{\partial \alpha_j} \mathbb{E}_q [\log \pi_k].$$

The matrix with entries

$$\frac{\partial}{\partial \alpha_j} \mathbb{E}_q \log \pi_k = \frac{\partial}{\partial \alpha_j} \{ \psi(\alpha_k) - \psi(\hat{\alpha}) \}$$

is the Hessian of the log normalizing function of the Dirichlet family. It is nonsingular on the interior of the simplex parameter space. Hence the stationary condition is

$$\alpha_k^* - \alpha_k = 0.$$

Therefore, the coordinate update is

$$\boxed{\alpha_k = \alpha_{0k} + N_k}.$$

In the symmetric prior case, $\alpha_{0k} = \alpha_0$, this becomes

$$\boxed{\alpha_k = \alpha_0 + N_k}.$$

Update for $q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$

For each component k , collect the terms depending on $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$. Up to terms not depending on $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$, we have

$$\begin{aligned} \log q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = & \frac{N_k}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \\ & + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{\beta_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \\ & + \frac{\nu_0 - D - 1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \boldsymbol{\Lambda}_k). \end{aligned}$$

Combine the log-determinant terms:

$$\frac{N_k}{2} + \frac{1}{2} + \frac{\nu_0 - D - 1}{2} = \frac{\nu_0 + N_k - D}{2}.$$

Since a Gaussian-Wishart density has log-determinant coefficient

$$\frac{\nu_k - D}{2},$$

we obtain by matching the coefficient after differentiation in the natural parameter associated with $\log |\boldsymbol{\Lambda}_k|$:

$$\boxed{\nu_k = \nu_0 + N_k}.$$

Next, collect the quadratic terms in $\boldsymbol{\mu}_k$:

$$\sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k) + \beta_0 (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0).$$

Differentiating this expression with respect to $\boldsymbol{\mu}_k$ gives

$$-2\boldsymbol{\Lambda}_k \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) - 2\beta_0 \boldsymbol{\Lambda}_k (\mathbf{m}_0 - \boldsymbol{\mu}_k).$$

Equivalently, after multiplying by $-1/2$, the derivative of the corresponding ELBO term is proportional to

$$\boldsymbol{\Lambda}_k \left\{ \sum_{n=1}^N r_{nk} \mathbf{x}_n + \beta_0 \mathbf{m}_0 - (N_k + \beta_0) \boldsymbol{\mu}_k \right\}.$$

Setting this derivative equal to zero yields

$$(N_k + \beta_0) \boldsymbol{\mu}_k = \sum_{n=1}^N r_{nk} \mathbf{x}_n + \beta_0 \mathbf{m}_0.$$

Therefore,

$$\boxed{\mathbf{m}_k = \frac{\beta_0 \mathbf{m}_0 + N_k \bar{\mathbf{x}}_k}{\beta_0 + N_k}}.$$

The corresponding posterior precision scaling for the conditional Gaussian factor is

$$\boxed{\beta_k = \beta_0 + N_k}.$$

To obtain $\mathbf{W}_k$, expand the quadratic terms around the updated mean. We have

$$\begin{aligned} & \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top + \beta_0 (\boldsymbol{\mu}_k - \mathbf{m}_0)(\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \\ = & N_k \mathbf{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k} (\bar{\mathbf{x}}_k - \mathbf{m}_0)(\bar{\mathbf{x}}_k - \mathbf{m}_0)^\top + \beta_k (\boldsymbol{\mu}_k - \mathbf{m}_k)(\boldsymbol{\mu}_k - \mathbf{m}_k)^\top. \end{aligned}$$

Therefore the terms involving $\boldsymbol{\Lambda}_k$ can be written as

$$\begin{aligned} \log q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = & \frac{\nu_k - D}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \text{tr} (\mathbf{W}_k^{-1} \boldsymbol{\Lambda}_k) \\ & - \frac{\beta_k}{2} (\boldsymbol{\mu}_k - \mathbf{m}_k)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_k) + \text{const}, \end{aligned}$$

where differentiation with respect to the matrix natural parameter gives

$$\boxed{\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + N_k \mathbf{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k} (\bar{\mathbf{x}}_k - \mathbf{m}_0)(\bar{\mathbf{x}}_k - \mathbf{m}_0)^\top}.$$

E Generalized vector autoregression model.

This section describes the Variational Bayes CAVI algorithm for the vector autoregressive model. Adapted from Roberts and Penny (2002) [9], which employs a univariate model.

E.1 The model and notation.

$$x_n = A_1 x_{n-1} + A_2 x_{n-2} + \cdots + A_p x_{n-p} + e_n, \quad n = p+1, \dots, N.$$

where x_n is a vector of D variables measure at n , A_i are $D \times D$ matrices. The disturbance term e_n is defined as:

$$\mathbf{e}_n \sim \sum_{k=1}^K \pi_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}).$$

Conditional on $z_{nk} = 1$,

$$\mathbf{e}_n \mid z_{nk} = 1 \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}).$$

Equivalently,

$$\mathbf{x}_n \mid z_{nk} = 1, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-p} \sim \mathcal{N} \left(\sum_{i=1}^p A_i \mathbf{x}_{n-i} + \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1} \right).$$

Note: The disturbances may be independent over time conditional on the latent allocations and the lagged observations, but their coordinates are not independent unless $\mathbf{\Lambda}_k$ is diagonal.

The latent allocation vector $\mathbf{z}_n$ is a 1-of- K binary vector, so that $z_{nk} \in \{0, 1\}$ and $\sum_{k=1}^K z_{nk} = 1$.

The mixing probabilities are denoted $\boldsymbol{\pi}$. Define the lag vector:

$$\mathbf{h}_n = \begin{pmatrix} \mathbf{x}_{n-1} \\ \mathbf{x}_{n-2} \\ \vdots \\ \mathbf{x}_{n-p} \end{pmatrix} \in \mathbb{R}^{Dp}.$$

Stack the autoregressive matrices into one coefficient matrix:

$$B = \begin{pmatrix} A_1^\top \\ A_2^\top \\ \vdots \\ A_p^\top \end{pmatrix} \in \mathbb{R}^{Dp \times D}.$$

Then

$$B^\top \mathbf{h}_n = \sum_{i=1}^p A_i \mathbf{x}_{n-i}.$$

The component-specific residual is given by:

$$\mathbf{r}_{nk} = \mathbf{x}_n - B^\top \mathbf{h}_n - \boldsymbol{\mu}_k.$$

The parameters of the model are denoted collectively as $\theta = \{\boldsymbol{\pi}, B, \boldsymbol{\mu}, \mathbf{\Lambda}\}$. When no component-specific subindex k is given, the parameters at all components are considered; e.g., $\boldsymbol{\mu}$ denotes $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$. The same applies to $\mathbf{\Lambda}$.

E.2 Complete-data log-likelihood function.

Conditional on the parameters θ the model factorizes as:

$$p(\mathbf{X}, \mathbf{Z} \mid \theta) = \prod_{n=p+1}^N p(\mathbf{z}_n \mid \boldsymbol{\pi}) p(x_n \mid \mathbf{z}_n, \mathbf{x}_n, \theta).$$

Conditional on the first p observations, the complete-data log-likelihood is:

$$\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\pi}, B, \boldsymbol{\mu}, \mathbf{\Lambda}) = \sum_{n=p+1}^N \sum_{k=1}^K z_{nk} \left[\log \pi_k + \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{1}{2} \mathbf{r}_{nk}^\top \mathbf{\Lambda}_k \mathbf{r}_{nk} \right].$$

E.3 Priors.

For a first variational extension of Roberts and Penny to the generalized VAR, we start with the direct analogue:

$$\text{vec}(B) \mid \alpha \sim \mathcal{N}\left(0, \alpha^{-1} I_{D^2 p}\right),$$

$$\alpha \sim \text{Gamma}(a_{\alpha 0}, r_{\alpha 0}).$$

This is comparable to the univariate prior.

$$B \mid \alpha \sim \mathcal{MN}_{Dp, D}\left(0, \alpha^{-1} R_0^{-1}, I_D\right),$$

where R_0 can encode lag-dependent shrinkage. For example, R_0 could be block diagonal:

$$R_0 = \text{blockdiag}(\kappa_1 I_D, \kappa_2 I_D, \dots, \kappa_p I_D),$$

with κ_i increasing with the lag order i , so that higher lags are shrunk more strongly.

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dirichlet}(\lambda_0, \dots, \lambda_0), \\ \text{vec}(B) \mid \alpha &\sim \mathcal{N}\left(0, \alpha^{-1} I_{D^2 p}\right), \\ \alpha &\sim \text{Gamma}(a_{\alpha 0}, r_{\alpha 0}), \\ \boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k &\sim \mathcal{N}\left(\mathbf{m}_0, (\kappa_0 \boldsymbol{\Lambda}_k)^{-1}\right), \\ \boldsymbol{\Lambda}_k &\sim \mathcal{W}(\mathbf{W}_0, \nu_0). \end{aligned}$$

In the univariate model of Roberts and Penny μ_s follows a Gaussian with precision v_0 , i.e. the precision does not depend on β there.

Full expression of $\log p(\theta)$:

$$\log p(\theta) = \log p(\boldsymbol{\pi}) + \log p(B \mid \alpha) + \log p(\alpha) + \sum_{k=1}^K \log p(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) + \sum_{k=1}^K \log p(\boldsymbol{\Lambda}_k),$$

with

$$\log p(\boldsymbol{\pi}) = \log \Gamma(K \lambda_0) - K \log \Gamma(\lambda_0) + \sum_{k=1}^K (\lambda_0 - 1) \log \pi_k.$$

$$\log p(B \mid \alpha) = \frac{M}{2} \log \alpha - \frac{M}{2} \log(2\pi) - \frac{\alpha}{2} \mathbf{b}^\top \mathbf{b}.$$

$$\log p(\alpha) = a_{\alpha 0} \log r_{\alpha 0} - \log \Gamma(a_{\alpha 0}) + (a_{\alpha 0} - 1) \log \alpha - r_{\alpha 0} \alpha.$$

$$\log p(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) = \frac{D}{2} \log \kappa_0 + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0).$$

$$\log p(\mathbf{\Lambda}_k) = \frac{\nu_0 - D - 1}{2} \log |\mathbf{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \mathbf{\Lambda}_k) - \frac{\nu_0 D}{2} \log 2 - \frac{\nu_0}{2} \log |\mathbf{W}_0| - \log \Gamma_D \left(\frac{\nu_0}{2} \right).$$

Thus:

$$\begin{aligned} \log p(\theta) = & \log \Gamma(K\lambda_0) - K \log \Gamma(\lambda_0) + \sum_{k=1}^K (\lambda_0 - 1) \log \pi_k \\ & + \frac{M}{2} \log \alpha - \frac{M}{2} \log(2\pi) - \frac{\alpha}{2} \mathbf{b}^\top \mathbf{b} \\ & + a_{\alpha 0} \log r_{\alpha 0} - \log \Gamma(a_{\alpha 0}) + (a_{\alpha 0} - 1) \log \alpha - r_{\alpha 0} \alpha \\ & + \sum_{k=1}^K \left[\frac{D}{2} \log \kappa_0 + \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \mathbf{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \right] \\ & + \sum_{k=1}^K \left[\frac{\nu_0 - D - 1}{2} \log |\mathbf{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \mathbf{\Lambda}_k) - \frac{\nu_0 D}{2} \log 2 - \frac{\nu_0}{2} \log |\mathbf{W}_0| - \log \Gamma_D \left(\frac{\nu_0}{2} \right) \right]. \end{aligned}$$

E.4 Variational densities.

The approximate posterior over the parameters is chosen by a mean-field factorization:

$$q(\theta, \mathbf{Z}) = q(\mathbf{Z})q(\boldsymbol{\pi})q(B)q(\alpha) \prod_{k=1}^K q(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k).$$

Here, the component mean and precision are treated as a joint Normal-Wishart factor $q(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k)$ rather than as two independent factors $q(\boldsymbol{\mu}_k)q(\mathbf{\Lambda}_k)$. This preserves conjugacy. Let's denote:

$$\mathbf{b} = \text{vec}(B), \quad \bar{\mathbf{b}} = \text{vec}(\bar{B}).$$

The factors have conjugate forms.

$$\begin{aligned} q(\mathbf{Z}) &= \prod_{n=p+1}^N \prod_{k=1}^K \gamma_{nk}^{z_{nk}}, \\ q(\boldsymbol{\pi}) &= \text{Dirichlet}(\lambda_1, \dots, \lambda_K), \\ q(B) &= \mathcal{N}(\mathbf{b} \mid \bar{\mathbf{b}}, \Sigma_B), \\ q(\alpha) &= \text{Gamma}(a_\alpha, r_\alpha), \\ q(\boldsymbol{\mu}_k, \mathbf{\Lambda}_k) &= q(\boldsymbol{\mu}_k \mid \mathbf{\Lambda}_k)q(\mathbf{\Lambda}_k), \quad \text{where} \\ q(\boldsymbol{\mu}_k \mid \mathbf{\Lambda}_k) &= \mathcal{N}(\mathbf{m}_k, (\kappa_k \mathbf{\Lambda}_k)^{-1}), \\ q(\mathbf{\Lambda}_k) &= \mathcal{W}(\mathbf{W}_k, \nu_k). \end{aligned}$$

Full expression of $\log q(\theta, \mathbf{Z})$:

$$\log q(\theta, \mathbf{Z}) = \log q(\mathbf{Z}) + \log q(\boldsymbol{\pi}) + \log q(B) + \log q(\alpha) + \sum_{k=1}^K \log q(\boldsymbol{\mu}_k \mid \mathbf{\Lambda}_k) + \sum_{k=1}^K \log q(\mathbf{\Lambda}_k),$$

with

$$\begin{aligned}\log q(\mathbf{Z}) &= \sum_{n=p+1}^N \sum_{k=1}^K z_{nk} \log \gamma_{nk} \\\log q(\boldsymbol{\pi}) &= \log \Gamma \left(\sum_{k=1}^K \lambda_k \right) - \sum_{k=1}^K \log \Gamma(\lambda_k) + \sum_{k=1}^K (\lambda_k - 1) \log \pi_k \\\log q(B) &= -\frac{M}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_B| - \frac{1}{2} (\mathbf{b} - \bar{\mathbf{b}})^\top \Sigma_B^{-1} (\mathbf{b} - \bar{\mathbf{b}}) \\\log q(\alpha) &= a_\alpha \log r_\alpha - \log \Gamma(a_\alpha) + (a_\alpha - 1) \log \alpha - r_\alpha \alpha.\end{aligned}$$

$$\log q(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) = \frac{D}{2} \log \kappa_k + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_k}{2} (\boldsymbol{\mu}_k - \mathbf{m}_k)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_k).$$

$$\log q(\boldsymbol{\Lambda}_k) = \frac{\nu_k - D - 1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_k^{-1} \boldsymbol{\Lambda}_k) - \frac{\nu_k D}{2} \log 2 - \frac{\nu_k}{2} \log |\mathbf{W}_k| - \log \Gamma_D \left(\frac{\nu_k}{2} \right).$$

Therefore:

$$\begin{aligned}\log q(\theta, \mathbf{Z}) &= \sum_{n=p+1}^N \sum_{k=1}^K z_{nk} \log \gamma_{nk} \\&+ \log \Gamma \left(\sum_{k=1}^K \lambda_k \right) - \sum_{k=1}^K \log \Gamma(\lambda_k) + \sum_{k=1}^K (\lambda_k - 1) \log \pi_k \\&- \frac{M}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_B| - \frac{1}{2} (\mathbf{b} - \bar{\mathbf{b}})^\top \Sigma_B^{-1} (\mathbf{b} - \bar{\mathbf{b}}) \\&+ a_\alpha \log r_\alpha - \log \Gamma(a_\alpha) + (a_\alpha - 1) \log \alpha - r_\alpha \alpha \\&+ \sum_{k=1}^K \left[\frac{D}{2} \log \kappa_k + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_k}{2} (\boldsymbol{\mu}_k - \mathbf{m}_k)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_k) \right] \\&+ \sum_{k=1}^K \left[\frac{\nu_k - D - 1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_k^{-1} \boldsymbol{\Lambda}_k) - \frac{\nu_k D}{2} \log 2 - \frac{\nu_k}{2} \log |\mathbf{W}_k| - \log \Gamma_D \left(\frac{\nu_k}{2} \right) \right].\end{aligned}$$

E.5 ELBO.

Auxiliary expectations for evaluating the ELBO.

The expression above keeps the factors visible. To evaluate it numerically, the following expectations are useful:

$$\mathbb{E}_q \log \pi_k = \psi(\lambda_k) - \psi \left(\sum_{j=1}^K \lambda_j \right).$$

$$\mathbb{E}_q \alpha = \frac{a_\alpha}{r_\alpha}.$$

$$\mathbb{E}_q \log \alpha = \psi(a_\alpha) - \log r_\alpha.$$

$$\mathbb{E}_q \log |\mathbf{\Lambda}_k| = \sum_{d=1}^D \psi \left(\frac{\nu_k + 1 - d}{2} \right) + D \log 2 + \log |\mathbf{W}_k|.$$

$$\mathbb{E}_q \mathbf{\Lambda}_k = \nu_k \mathbf{W}_k.$$

$$\mathbb{E}_q \mathbf{b}^\top \mathbf{b} = \bar{\mathbf{b}}^\top \bar{\mathbf{b}} + \text{tr}(\Sigma_B).$$

The multivariate gamma function is:

$$\Gamma_D(a) = \pi^{D(D-1)/4} \prod_{d=1}^D \Gamma \left(a + \frac{1-d}{2} \right).$$

Upon the definitions and expressions given above, the ELBO is given by:

$$\text{ELBO}(q) = \mathbb{E}_{q(\theta)q(\mathbf{Z})} \log p(\mathbf{X}, \mathbf{Z} \mid \theta) + \mathbb{E}_{q(\theta)} [\log p(\theta)] - \mathbb{E}_{q(\theta)} [\log q(\theta)] - \mathbb{E}_{q(\mathbf{Z})} [\log q(\mathbf{Z})].$$

Taking expectations on q of the random variables, we get:

$$\begin{aligned} \text{ELBO}(q) = & \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \left[\mathbb{E}_q \log \pi_k + \frac{1}{2} \mathbb{E}_q [\log |\mathbf{\Lambda}_k|] - \frac{D}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_q [\mathbf{r}_{nk}^\top \mathbf{\Lambda}_k \mathbf{r}_{nk}] \right] \\ & + \mathbb{E}_q \left[\log \Gamma(K\lambda_0) - K \log \Gamma(\lambda_0) + \sum_{k=1}^K (\lambda_0 - 1) \log \pi_k \right] \\ & + \mathbb{E}_q \left[\frac{M}{2} \log \alpha - \frac{M}{2} \log(2\pi) - \frac{\alpha}{2} \mathbf{b}^\top \mathbf{b} \right] \\ & + \mathbb{E}_q [a_{\alpha 0} \log r_{\alpha 0} - \log \Gamma(a_{\alpha 0}) + (a_{\alpha 0} - 1) \log \alpha - r_{\alpha 0} \alpha] \\ & + \sum_{k=1}^K \mathbb{E}_q \left[\frac{D}{2} \log \kappa_0 + \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \mathbf{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \right] \\ & + \sum_{k=1}^K \mathbb{E}_q \left[\frac{\nu_0 - D - 1}{2} \log |\mathbf{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \mathbf{\Lambda}_k) - \frac{\nu_0 D}{2} \log 2 - \frac{\nu_0}{2} \log |\mathbf{W}_0| - \log \Gamma_D \left(\frac{\nu_0}{2} \right) \right] \\ & - \mathbb{E}_q \left[\log \Gamma \left(\sum_{k=1}^K \lambda_k \right) - \sum_{k=1}^K \log \Gamma(\lambda_k) + \sum_{k=1}^K (\lambda_k - 1) \log \pi_k \right] \\ & - \mathbb{E}_q \left[-\frac{M}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_B| - \frac{1}{2} (\mathbf{b} - \bar{\mathbf{b}})^\top \Sigma_B^{-1} (\mathbf{b} - \bar{\mathbf{b}}) \right] \\ & - \mathbb{E}_q [a_\alpha \log r_\alpha - \log \Gamma(a_\alpha) + (a_\alpha - 1) \log \alpha - r_\alpha \alpha] \\ & - \sum_{k=1}^K \mathbb{E}_q \left[\frac{D}{2} \log \kappa_k + \frac{1}{2} \log |\mathbf{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_k}{2} (\boldsymbol{\mu}_k - \mathbf{m}_k)^\top \mathbf{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_k) \right] \\ & - \sum_{k=1}^K \mathbb{E}_q \left[\frac{\nu_k - D - 1}{2} \log |\mathbf{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_k^{-1} \mathbf{\Lambda}_k) - \frac{\nu_k D}{2} \log 2 - \frac{\nu_k}{2} \log |\mathbf{W}_k| - \log \Gamma_D \left(\frac{\nu_k}{2} \right) \right] \\ & - \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \log \gamma_{nk}. \end{aligned}$$

Let

$$M = D^2 p, \quad \mathbf{b} = \text{vec}(B), \quad \bar{\mathbf{b}} = \text{vec}(\bar{B}).$$

For each observation n , define the matrix G_n such that

$$B^\top \mathbf{h}_n = G_n \mathbf{b},$$

where

$$G_n = I_D \otimes \mathbf{h}_n^\top.$$

Define also

$$\ell_{\pi k} = \mathbb{E}_q \log \pi_k = \psi(\lambda_k) - \psi\left(\sum_{j=1}^K \lambda_j\right),$$

$$\ell_\alpha = \mathbb{E}_q \log \alpha = \psi(a_\alpha) - \log r_\alpha,$$

$$\bar{\alpha} = \mathbb{E}_q[\alpha] = \frac{a_\alpha}{r_\alpha},$$

$$\ell_{\Lambda k} = \mathbb{E}_q[\log |\Lambda_k|] = \sum_{d=1}^D \psi\left(\frac{\nu_k + 1 - d}{2}\right) + D \log 2 + \log |\mathbf{W}_k|,$$

$$\bar{\Lambda}_k = \mathbb{E}_q[\Lambda_k] = \nu_k \mathbf{W}_k,$$

$$\mathbb{E}_q[\mathbf{b}^\top \mathbf{b}] = \bar{\mathbf{b}}^\top \bar{\mathbf{b}} + \text{tr}(\Sigma_B).$$

The expected quadratic likelihood term can be written as

$$Q_{nk} = \mathbb{E}_q \left[\mathbf{r}_{nk}^\top \Lambda_k \mathbf{r}_{nk} \right],$$

where

$$\mathbf{r}_{nk} = \mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k.$$

Using the moments of $q(B)$ and the Normal-Wishart factor $q(\boldsymbol{\mu}_k, \Lambda_k)$, this becomes

$$Q_{nk} = \boldsymbol{\delta}_{nk}^\top \bar{\Lambda}_k \boldsymbol{\delta}_{nk} + \text{tr} \left(\bar{\Lambda}_k G_n \Sigma_B G_n^\top \right) + \frac{D}{\kappa_k},$$

with

$$\boldsymbol{\delta}_{nk} = \mathbf{x}_n - G_n \bar{\mathbf{b}} - \mathbf{m}_k.$$

Similarly, the prior quadratic term for the component mean is

$$Q_{\mu k}^{(0)} = \mathbb{E}_q \left[(\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \Lambda_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \right],$$

so that

$$Q_{\mu k}^{(0)} = (\mathbf{m}_k - \mathbf{m}_0)^\top \bar{\Lambda}_k (\mathbf{m}_k - \mathbf{m}_0) + \frac{D}{\kappa_k}.$$

Thus,

$$\begin{aligned}
 \text{ELBO}(q) = & \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \left[\ell_{\pi k} + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) \right. \\
 & \left. - \frac{1}{2} \left\{ \boldsymbol{\delta}_{nk}^\top \bar{\boldsymbol{\Lambda}}_k \boldsymbol{\delta}_{nk} + \text{tr}(\bar{\boldsymbol{\Lambda}}_k G_n \Sigma_B G_n^\top) + \frac{D}{\kappa_k} \right\} \right] \\
 & + \left[\log \Gamma(K \lambda_0) - K \log \Gamma(\lambda_0) + \sum_{k=1}^K (\lambda_0 - 1) \ell_{\pi k} \right] \\
 & + \left[\frac{M}{2} \ell_\alpha - \frac{M}{2} \log(2\pi) - \frac{\bar{\alpha}}{2} (\bar{\mathbf{b}}^\top \bar{\mathbf{b}} + \text{tr}(\Sigma_B)) \right] \\
 & + [a_{\alpha 0} \log r_{\alpha 0} - \log \Gamma(a_{\alpha 0}) + (a_{\alpha 0} - 1) \ell_\alpha - r_{\alpha 0} \bar{\alpha}] \\
 & + \sum_{k=1}^K \left[\frac{D}{2} \log \kappa_0 + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) \right. \\
 & \left. - \frac{\kappa_0}{2} \left\{ (\mathbf{m}_k - \mathbf{m}_0)^\top \bar{\boldsymbol{\Lambda}}_k (\mathbf{m}_k - \mathbf{m}_0) + \frac{D}{\kappa_k} \right\} \right] \\
 & + \sum_{k=1}^K \left[\frac{\nu_0 - D - 1}{2} \ell_{\Lambda k} - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \bar{\boldsymbol{\Lambda}}_k) - \frac{\nu_0 D}{2} \log 2 - \frac{\nu_0}{2} \log |\mathbf{W}_0| - \log \Gamma_D \left(\frac{\nu_0}{2} \right) \right] \\
 & - \left[\log \Gamma \left(\sum_{k=1}^K \lambda_k \right) - \sum_{k=1}^K \log \Gamma(\lambda_k) + \sum_{k=1}^K (\lambda_k - 1) \ell_{\pi k} \right] \\
 & - \left[-\frac{M}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_B| - \frac{M}{2} \right] \\
 & - [a_\alpha \log r_\alpha - \log \Gamma(a_\alpha) + (a_\alpha - 1) \ell_\alpha - r_\alpha \bar{\alpha}] \\
 & - \sum_{k=1}^K \left[\frac{D}{2} \log \kappa_k + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) - \frac{D}{2} \right] \\
 & - \sum_{k=1}^K \left[\frac{\nu_k - D - 1}{2} \ell_{\Lambda k} - \frac{1}{2} \text{tr}(\mathbf{W}_k^{-1} \bar{\boldsymbol{\Lambda}}_k) - \frac{\nu_k D}{2} \log 2 - \frac{\nu_k}{2} \log |\mathbf{W}_k| - \log \Gamma_D \left(\frac{\nu_k}{2} \right) \right] \\
 & - \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \log \gamma_{nk}.
 \end{aligned}$$

E.6 Coordinate-ascent algorithm.

Update for $q(\mathbf{Z})$: responsibilities γ_{nk} .

The terms of the ELBO that depend on $q(\mathbf{Z})$ are:

$$\mathbb{E}_{q(\theta)q(\mathbf{Z})} \log p(\mathbf{X}, \mathbf{Z} \mid \theta) - \mathbb{E}_{q(\mathbf{Z})} \log q(\mathbf{Z}).$$

Keeping only the terms involving γ_{nk} ,

$$\mathcal{L}_\gamma = \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \left[\ell_{\pi k} + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk} \right] - \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \log \gamma_{nk},$$

where

$$Q_{nk} = \mathbb{E}_q \left[\mathbf{r}_{nk}^\top \mathbf{\Lambda}_k \mathbf{r}_{nk} \right].$$

Using the current variational moments,

$$Q_{nk} = \boldsymbol{\delta}_{nk}^\top \bar{\mathbf{\Lambda}}_k \boldsymbol{\delta}_{nk} + \text{tr} \left(\bar{\mathbf{\Lambda}}_k G_n \Sigma_B G_n^\top \right) + \frac{D}{\kappa_k},$$

with

$$\boldsymbol{\delta}_{nk} = \mathbf{x}_n - G_n \bar{\mathbf{b}} - \mathbf{m}_k.$$

Impose the constraint

$$\sum_{k=1}^K \gamma_{nk} = 1.$$

The Lagrangian for each n is

$$\mathcal{J}_{\gamma_n} = \sum_{k=1}^K \gamma_{nk} \left[\ell_{\pi k} + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk} \right] - \sum_{k=1}^K \gamma_{nk} \log \gamma_{nk} + \rho_n \left(\sum_{k=1}^K \gamma_{nk} - 1 \right).$$

The derivative is

$$\frac{\partial \mathcal{J}_{\gamma_n}}{\partial \gamma_{nk}} = \ell_{\pi k} + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk} - \log \gamma_{nk} - 1 + \rho_n.$$

Equating to zero gives

$$\log \gamma_{nk} = \ell_{\pi k} + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk} + \rho_n - 1.$$

Therefore, after normalization,

$$\gamma_{nk} = \frac{\exp(a_{nk})}{\sum_{j=1}^K \exp(a_{nj})},$$

where

$$a_{nk} = \ell_{\pi k} + \frac{1}{2} \ell_{\Lambda k} - \frac{D}{2} \log(2\pi) - \frac{1}{2} Q_{nk}.$$

Since the term $-D \log(2\pi)/2$ is common to all k , it may be omitted in the numerical softmax.

Update for $q(\boldsymbol{\pi})$: Dirichlet parameters λ_k .

The terms of the ELBO that depend on $\boldsymbol{\pi}$ are:

$$\mathbb{E}_{q(\theta)q(\mathbf{Z})} \log p(\mathbf{Z} \mid \boldsymbol{\pi}) + \mathbb{E}_{q(\theta)} \log p(\boldsymbol{\pi}) - \mathbb{E}_{q(\theta)} \log q(\boldsymbol{\pi}).$$

The terms involving $\boldsymbol{\pi}$ are:

$$\sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \log \pi_k + \sum_{k=1}^K (\lambda_0 - 1) \log \pi_k.$$

Collecting terms,

$$\log q^*(\boldsymbol{\pi}) = \sum_{k=1}^K \left(\lambda_0 + \sum_{n=p+1}^N \gamma_{nk} - 1 \right) \log \pi_k + \text{constant}.$$

This has the Dirichlet form. Define

$$N_k = \sum_{n=p+1}^N \gamma_{nk}.$$

Then

$$q^*(\boldsymbol{\pi}) = \text{Dirichlet}(\lambda_1, \dots, \lambda_K),$$

with update

$$\lambda_k = \lambda_0 + N_k.$$

Equivalently, one can see this by differentiating the relevant ELBO with respect to the Dirichlet natural coefficient of $\log \pi_k$. The updated coefficient is the prior coefficient plus the expected number of allocations to component k :

$$\lambda_k - 1 = (\lambda_0 - 1) + N_k.$$

Therefore,

$$\boxed{\lambda_k = \lambda_0 + \sum_{n=p+1}^N \gamma_{nk}}.$$

Update for $q(B)$: Gaussian factor for $\mathbf{b} = \text{vec}(B)$.

The terms of the ELBO that depend on B are the expected complete-data likelihood and the Gaussian prior $p(B \mid \alpha)$:

$$\mathbb{E}_{q(\theta)q(\mathbf{Z})} \log p(\mathbf{X}, \mathbf{Z} \mid \theta) + \mathbb{E}_{q(\theta)} \log p(B \mid \alpha) - \mathbb{E}_{q(\theta)} \log q(B).$$

Keeping only the terms involving $\mathbf{b}$,

$$\mathcal{L}_B = -\frac{1}{2} \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} \mathbb{E}_q \left[(\mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k) \right] - \frac{1}{2} \bar{\alpha} \mathbf{b}^\top \mathbf{b} + \text{entropy of } q(B).$$

For the coordinate update, collect the quadratic terms in $\mathbf{b}$. Using

$$\mathbb{E}_q \boldsymbol{\mu}_k = \mathbf{m}_k, \quad \mathbb{E}_q \boldsymbol{\Lambda}_k = \bar{\boldsymbol{\Lambda}}_k,$$

the $\mathbf{b}$ -dependent part is

$$\mathcal{L}_B = -\frac{1}{2} \mathbf{b}^\top \left[\bar{\alpha} I_M + \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} G_n^\top \bar{\boldsymbol{\Lambda}}_k G_n \right] \mathbf{b} + \mathbf{b}^\top \left[\sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} G_n^\top \bar{\boldsymbol{\Lambda}}_k (\mathbf{x}_n - \mathbf{m}_k) \right] + \text{constant}.$$

Define

$$\Sigma_B^{-1} = \bar{\alpha} I_M + \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} G_n^\top \bar{\Lambda}_k G_n.$$

Define also

$$\mathbf{s}_B = \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} G_n^\top \bar{\Lambda}_k (\mathbf{x}_n - \mathbf{m}_k).$$

The derivative with respect to $\mathbf{b}$ is

$$\frac{\partial \mathcal{L}_B}{\partial \mathbf{b}} = -\Sigma_B^{-1} \mathbf{b} + \mathbf{s}_B.$$

Equating to zero gives

$$\Sigma_B^{-1} \bar{\mathbf{b}} = \mathbf{s}_B.$$

Therefore,

$$\Sigma_B = \left[\bar{\alpha} I_M + \sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} G_n^\top \bar{\Lambda}_k G_n \right]^{-1}$$

and

$$\bar{\mathbf{b}} = \Sigma_B \left[\sum_{n=p+1}^N \sum_{k=1}^K \gamma_{nk} G_n^\top \bar{\Lambda}_k (\mathbf{x}_n - \mathbf{m}_k) \right].$$

Finally, reshape

$$\bar{B} = \text{matrix}(\bar{\mathbf{b}}, Dp, D).$$

Update for $q(\alpha)$: Gamma parameters a_α, r_α .

The terms of the ELBO that depend on α are:

$$\mathbb{E}_{q(\theta)} \log p(B \mid \alpha) + \mathbb{E}_{q(\theta)} \log p(\alpha) - \mathbb{E}_{q(\theta)} \log q(\alpha).$$

The α -dependent prior terms are:

$$\log p(B \mid \alpha) = \frac{M}{2} \log \alpha - \frac{M}{2} \log(2\pi) - \frac{\alpha}{2} \mathbf{b}^\top \mathbf{b},$$

and

$$\log p(\alpha) = a_{\alpha 0} \log r_{\alpha 0} - \log \Gamma(a_{\alpha 0}) + (a_{\alpha 0} - 1) \log \alpha - r_{\alpha 0} \alpha.$$

Keeping only terms involving α ,

$$\log q^*(\alpha) = \left(a_{\alpha 0} + \frac{M}{2} - 1 \right) \log \alpha - \left[r_{\alpha 0} + \frac{1}{2} \mathbb{E}_q(\mathbf{b}^\top \mathbf{b}) \right] \alpha + \text{constant}.$$

This is a Gamma density. Therefore,

$$q^*(\alpha) = \text{Gamma}(a_\alpha, r_\alpha),$$

with

$$a_\alpha = a_{\alpha 0} + \frac{M}{2},$$

and

$$r_\alpha = r_{\alpha 0} + \frac{1}{2} \mathbb{E}_q[\mathbf{b}^\top \mathbf{b}].$$

Using the Gaussian moment,

$$\mathbb{E}_q[\mathbf{b}^\top \mathbf{b}] = \bar{\mathbf{b}}^\top \bar{\mathbf{b}} + \text{tr}(\Sigma_B).$$

Hence,

$$a_\alpha = a_{\alpha 0} + \frac{D^2 p}{2}$$

and

$$r_\alpha = r_{\alpha 0} + \frac{1}{2} [\bar{\mathbf{b}}^\top \bar{\mathbf{b}} + \text{tr}(\Sigma_B)].$$

The corresponding expectation used in other updates is

$$\bar{\alpha} = \frac{a_\alpha}{r_\alpha}.$$

Update for $q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$: Normal-Wishart parameters.

The terms of the ELBO that depend on $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ are:

$$\mathbb{E}_{q(\theta)q(\mathbf{Z})} \log p(\mathbf{X}, \mathbf{Z} \mid \theta) + \mathbb{E}_{q(\theta)} \log p(\boldsymbol{\mu}_k \mid \boldsymbol{\Lambda}_k) + \mathbb{E}_{q(\theta)} \log p(\boldsymbol{\Lambda}_k) - \mathbb{E}_{q(\theta)} \log q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k).$$

For fixed k , keep only terms involving $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$. Let

$$\bar{\mathbf{e}}_n = \mathbf{x}_n - G_n \bar{\mathbf{b}},$$

and

$$C_n = G_n \Sigma_B G_n^\top.$$

The likelihood contribution contains

$$\sum_{n=p+1}^N \gamma_{nk} \left[\frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \mathbb{E}_{q(B)} \left[(\mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k) \right] \right].$$

Since

$$\mathbb{E}_{q(B)} \left[(\mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k) (\mathbf{x}_n - G_n \mathbf{b} - \boldsymbol{\mu}_k)^\top \right] = (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k) (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k)^\top + C_n,$$

the relevant terms are

$$\begin{aligned} \log q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) &= \sum_{n=p+1}^N \gamma_{nk} \left[\frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k) - \frac{1}{2} \text{tr}(\boldsymbol{\Lambda}_k C_n) \right] \\ &\quad + \frac{D}{2} \log \kappa_0 + \frac{1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{D}{2} \log(2\pi) - \frac{\kappa_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0) \\ &\quad + \frac{\nu_0 - D - 1}{2} \log |\boldsymbol{\Lambda}_k| - \frac{1}{2} \text{tr}(\mathbf{W}_0^{-1} \boldsymbol{\Lambda}_k) + \text{constant}. \end{aligned}$$

Define the effective component size

$$N_k = \sum_{n=p+1}^N \gamma_{nk}.$$

Define the weighted residual mean

$$\bar{\mathbf{e}}_k = \frac{1}{N_k} \sum_{n=p+1}^N \gamma_{nk} \bar{\mathbf{e}}_n.$$

Define the weighted scatter matrix including uncertainty from $q(B)$:

$$S_k = \sum_{n=p+1}^N \gamma_{nk} \left[(\bar{\mathbf{e}}_n - \bar{\mathbf{e}}_k)(\bar{\mathbf{e}}_n - \bar{\mathbf{e}}_k)^\top + C_n \right].$$

Update for $\mathbf{m}_k$

The terms depending on $\boldsymbol{\mu}_k$, conditional on $\boldsymbol{\Lambda}_k$, are proportional to

$$-\frac{1}{2} \sum_{n=p+1}^N \gamma_{nk} (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k) - \frac{\kappa_0}{2} (\boldsymbol{\mu}_k - \mathbf{m}_0)^\top \boldsymbol{\Lambda}_k (\boldsymbol{\mu}_k - \mathbf{m}_0).$$

The derivative with respect to $\boldsymbol{\mu}_k$ is

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} = \boldsymbol{\Lambda}_k \left[\sum_{n=p+1}^N \gamma_{nk} (\bar{\mathbf{e}}_n - \boldsymbol{\mu}_k) + \kappa_0 (\mathbf{m}_0 - \boldsymbol{\mu}_k) \right].$$

Equating to zero gives

$$\sum_{n=p+1}^N \gamma_{nk} \bar{\mathbf{e}}_n + \kappa_0 \mathbf{m}_0 - (N_k + \kappa_0) \boldsymbol{\mu}_k = 0.$$

Therefore,

$$\mathbf{m}_k = \frac{\kappa_0 \mathbf{m}_0 + N_k \bar{\mathbf{e}}_k}{\kappa_0 + N_k}.$$

Update for κ_k

The posterior conditional covariance of $\boldsymbol{\mu}_k$ is determined by the coefficient multiplying $\boldsymbol{\Lambda}_k$ in the quadratic term for $\boldsymbol{\mu}_k$. The likelihood contributes $N_k \boldsymbol{\Lambda}_k$, and the prior contributes $\kappa_0 \boldsymbol{\Lambda}_k$. Hence,

$$\kappa_k = \kappa_0 + N_k.$$

Update for ν_k

The coefficient of $\log |\mathbf{\Lambda}_k|$ is obtained by adding the likelihood, Normal prior, and Wishart prior contributions:

$$\frac{N_k}{2} + \frac{1}{2} + \frac{\nu_0 - D - 1}{2} = \frac{\nu_0 + N_k - D}{2}.$$

The Wishart variational density has coefficient

$$\frac{\nu_k - D - 1}{2}$$

inside $q(\mathbf{\Lambda}_k)$, while the Normal conditional $q(\boldsymbol{\mu}_k \mid \mathbf{\Lambda}_k)$ contributes an additional $1/2 \log |\mathbf{\Lambda}_k|$. Together, the Normal-Wishart form implies

$$\boxed{\nu_k = \nu_0 + N_k.}$$

Update for $\mathbf{W}_k$

After completing the square in $\boldsymbol{\mu}_k$, the matrix multiplying $\mathbf{\Lambda}_k$ in the trace term is

$$\mathbf{W}_0^{-1} + S_k + \frac{\kappa_0 N_k}{\kappa_0 + N_k} (\bar{\mathbf{e}}_k - \mathbf{m}_0)(\bar{\mathbf{e}}_k - \mathbf{m}_0)^\top.$$

The Wishart variational factor has trace term

$$-\frac{1}{2} \text{tr}(\mathbf{W}_k^{-1} \mathbf{\Lambda}_k).$$

Therefore, matching the trace coefficient gives

$$\boxed{\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + S_k + \frac{\kappa_0 N_k}{\kappa_0 + N_k} (\bar{\mathbf{e}}_k - \mathbf{m}_0)(\bar{\mathbf{e}}_k - \mathbf{m}_0)^\top.}$$

Equivalently,

$$\boxed{\mathbf{W}_k = \left[\mathbf{W}_0^{-1} + S_k + \frac{\kappa_0 N_k}{\kappa_0 + N_k} (\bar{\mathbf{e}}_k - \mathbf{m}_0)(\bar{\mathbf{e}}_k - \mathbf{m}_0)^\top \right]^{-1}.}$$

The full Normal-Wishart update is therefore:

$$\boxed{\kappa_k = \kappa_0 + N_k, \quad \nu_k = \nu_0 + N_k,}$$

$$\boxed{\mathbf{m}_k = \frac{\kappa_0 \mathbf{m}_0 + N_k \bar{\mathbf{e}}_k}{\kappa_0 + N_k},}$$

$$\boxed{\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + S_k + \frac{\kappa_0 N_k}{\kappa_0 + N_k} (\bar{\mathbf{e}}_k - \mathbf{m}_0)(\bar{\mathbf{e}}_k - \mathbf{m}_0)^\top.}$$