Quantized Matrix Completion through Bilinear Factorization and Approximate Message Passing

by

Anis Housseini

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg

Acknowledgement

By way of recognition, I would like to express my sincerest thanks to everyone who contributed to the success of this project and the development of this work.

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Amine Mezghani, for his invaluable guidance, unwavering moral support, and patience in guiding me through all the stages of my research. I also want to warmly thank Prof. Faouzi Bellili, my co-advisor, for his fruitful advice and his availability to answer the various questions I asked in order to create an efficient and effective tool suited to their specific needs.

I would also like to extend my special thanks to Mohamed AKROUT, a Ph.D. student at the University of Manitoba, for his grateful effort, encouragement, and guidance throughout my research period.

My final heartfelt thanks to my beloved family, and friends, for their support, and the entire research team for sharing information, knowledge, and collaboration during this research stay.

Abstract

Compressed sensing is an exciting, quickly developing field, pulling in significant consideration in electrical engineering, applied mathematics, statistics, and software engineering. It can likewise serve computer vision, coding theory, signal processing, image processing, and algorithms for efficient data processing. Compressed sensing gives an exquisite framework for recuperating signals from compressed measurements. In this regard, our main focus in this thesis is the problem of bilinear factorization of sub-sampled matrices, also known as matrix factorization (MF), which has different variations, such as unconstrained MF, integer MF (IMF), non-negative MF (NMF), probabilistic MF (PMF), to cite a few. In this thesis, we introduce a new Bayesian quantized MF solution based on the approximate message passing (AMP) framework to solve the discrete matrix completion (MC) problem. Specifically, we resort to the bilinear generalized vector AMP (BiG-VAMP) algorithm along with a quantizer, and an expectation-maximization (EM) learning procedure to optimize the quantization thresholds. This approach enables our algorithm to jointly recover two matrices, denoted as $\boldsymbol{U}$ and $\boldsymbol{V}$, as well as their real-valued and quantized product matrices from an incomplete discrete low-rank matrix $\boldsymbol{Y}$, observed through a component-wise selection transformation. Extensive computer simulations, based on synthetic and real-world discrete datasets, show that the proposed method outperforms the state-of-the-art techniques for MF with discrete-valued factors in terms of reconstruction performance. Moreover, we conduct a theoretical performance analysis of the proposed method, based on the state evolution (SE) framework, and show its consistency with the empirical MSE performance obtained by computer simulations.

Contents

List of Figures

List of Algorithms

Notation

Throughout this report, the following notations will be used in mathematical equations unless otherwise stated:

- We use x, $\mathbf{x}$, and $\mathbf{X}$ to denote random variables, vectors, and matrices while x, $\boldsymbol{x}$, and $\boldsymbol{X}$ denote their realizations, respectively.
- Given any matrix $\boldsymbol{X}$, we use $\boldsymbol{x}_i$ and x_{ij} to denote its ith column and ijth entry, respectively. The operator $\mathrm{diag}(\boldsymbol{X})$ stacks the diagonal elements of $\boldsymbol{X}$ in a vector while $\boldsymbol{I}$ stands for the identity matrix. The operator $\mathrm{tr}(\boldsymbol{X})$ returns the trace of the matrix $\boldsymbol{X}$.
- We use $p_{\mathsf{X}}(x)$ to denote the pdf of any random variable X. Moreover, $\mathcal{N}(x; \widehat{x}, \sigma_x^2)$ (resp., $\mathcal{N}(\boldsymbol{X}; \widehat{\boldsymbol{X}}, \boldsymbol{\Sigma})$) stands for the Gaussian pdf of a scalar (resp., multivariate) random variable x (resp., X) with mean $\widehat{x}$ (resp., $\widehat{\boldsymbol{X}}$) and variance σ_x^2 (resp., covariance matrix $\boldsymbol{\Sigma}$). We write $\sim$ and $\propto$ to say "distributed according to" and "proportional to", respectively.
- We also use $\mathbb{E}_{\mathbf{X}}\{.\}$ (resp., $\mathbb{E}_{\mathbf{X};\boldsymbol{\theta}}\{.\}$) to denote the statistical expectation of $\mathbf{X}$ (resp., $\mathbf{X}$ as being parameterized by $\boldsymbol{\theta}$) and $\delta(\boldsymbol{x})$ refers to the Dirac delta distribution. We denote by $\langle \boldsymbol{x} \rangle$ and $\langle \boldsymbol{X} \rangle$ the empirical vector and matrix average value, i.e., $\langle \boldsymbol{x} \rangle \triangleq \frac{1}{N} \sum_{i=1}^{N} x_i$ for $\boldsymbol{x} \in \mathbb{R}^N$, and $\langle \boldsymbol{X} \rangle \triangleq \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij}$ for $\boldsymbol{X} \in \mathbb{R}^{N \times M}$.
- The symbol $\odot$ stands for the Hadamard product between any two matrices.
- Finally, $\Phi(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} \, \mathrm{d}t$ denotes the cumulative Gaussian distribution function.

Introduction

The real-valued matrix completion (MC) problem considers a partially observed low-rank matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$ obtained from the following generalized bilinear model:

$$\boldsymbol{Y} = \phi_s(\boldsymbol{Z}) \;\text{ with }\; \boldsymbol{Z} = \boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}, \tag{1}$$

Here, $\boldsymbol{U} \in \mathbb{R}^{N \times R}$ and $\boldsymbol{V} \in \mathbb{R}^{M \times R}$ are the unknown row-factor and column-factor matrices over a latent factor of size R, with $R \ll N, M$. Moreover, $\phi_s(\cdot)$ is a selection channel representing the process by which a subset Ω of the entries of $\boldsymbol{Y}$ is unknown (e.g., equal to 0). That is to say:

$$\boldsymbol{Y} = \phi_s(\boldsymbol{Z}) = \boldsymbol{S} \odot \boldsymbol{Z}, \tag{2}$$

where $\boldsymbol{S}$ is selection matrix whose entry s_{ij} is given by:

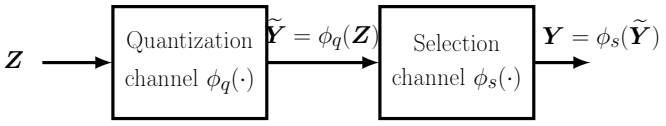
$$s_{ij} = \begin{cases} 1\,, & \text{if } \; y_{ij} \text{ is known} \\ 0\,, & \text{otherwise} \end{cases} \tag{3}$$

Finding an efficient algorithm for recovering lost information of a full matrix from only a known subset of its entries Ω is a recurring problem in a myriad of practical applications such as recommendation systems (RSs) [50], [51], social/biological network analysis [47], [55], massive multiple-input multiple-output (MIMO) in wireless communication [48], [49], internet of things (IoT) sensor networks [52], and system identification [53], to cite a few.

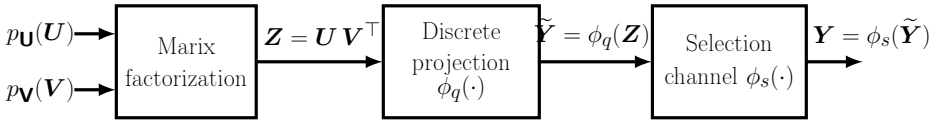
In these aforementioned applications, however, the entries of $\boldsymbol{Y}$ are not always real-valued but rather discrete-valued such as the case for recommendation systems where the observed matrix has integers between 1 and 5 reflecting user ratings of certain movies. It is also possible to impose discrete constraints on the factors $\boldsymbol{U}$ and/or $\boldsymbol{V}$ so as to enforce prior knowledge about the process that generates the observed data. This can help to improve the ability to detect structures

in the observed data. For instance, in the RSs example, instead of using the widely adopted Gaussian prior in the RSs applications, a binary prior on $\boldsymbol{U}$ may reflect that a user either likes or dislikes a movie category. Similarly, binary-valued factors are the appropriate choice when exclusive choices are justified, such as in the active/inactive pathway in genomics or the connected/disconnected device status in wireless networks. [16].

To enforce discrete prior knowledge about $\boldsymbol{Y}$, $\boldsymbol{U}$, and/or $\boldsymbol{V}$, the MC problem can be regarded as a special case of Bayesian bilinear recovery problems. In this case, the matrix factorization (MF) problem involves two priors $p_{\mathbf{U}}(\boldsymbol{U})$ and $p_{\mathbf{V}}(\boldsymbol{V})$ of the latent factors $\boldsymbol{U}$ and $\boldsymbol{V}$. The MF is then followed by a quantization channel $\phi_q(\cdot)$ and a selection channel $\phi_s(\cdot)$ as depicted in Fig. 1b.



(a) Matrix completion



(b) Matrix completion through bilinear factorization

Figure 1: The quantized MC transformation modeled as a cascade of a quantization channel $\phi_q(\cdot)$ followed by a selection channel $\phi_s(\cdot)$. The goal is to recover in (a) a matrix $\boldsymbol{Z}$ or in (b) two matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ through the explicit bilinear factorization $\boldsymbol{Z} = \boldsymbol{U}\,\boldsymbol{V}^\top$.

The observation model of $\boldsymbol{Y}$ given in (1) is then expressed as:

$$\boldsymbol{Y} = \phi_s\left(\widetilde{\boldsymbol{Y}}\right) = \phi_s\left(\phi_q\left(\boldsymbol{U}\,\boldsymbol{V}^\top\right)\right) = \boldsymbol{S} \odot \phi_q\left(\boldsymbol{U}\,\boldsymbol{V}^\top\right), \tag{4}$$

where $\widetilde{\boldsymbol{Y}} \triangleq \phi_q\left(\boldsymbol{U}\,\boldsymbol{V}^\top\right)$ is the quantized matrix before the selection channel, whose entries belong to a l-discrete alphabet, $\boldsymbol{e} = \{e_1, e_2, \ldots, e_l\}$ with $l \ll NM$.

The quantizer $\phi_q(\cdot)$ maps a real-valued input to a finite set of l-discrete values defined according to the quantization thresholds $\boldsymbol{r} \triangleq [r_1, r_2, \ldots, r_{l-1}]^\top$. By defining $r_0 = -\infty$ and $r_l = +\infty$ such that $r_0 < r_1 < r_2 < \cdots < r_{l-1} < r_l$,

the quantization output is assigned to one of the l discrete values determined as follows:

$$\widetilde{y}_{ij} = \phi_q(z_{ij}) = \begin{cases} e_1\,, & z_{ij} \in \,]r_0, r_1] \\ e_2\,, & z_{ij} \in \,]r_1, r_2] \\ & \vdots \\ e_l\,, & z_{ij} \in \,[r_{l-1}, r_l[\end{cases} \tag{5}$$

The MC formulation in (4) handles two types of discrete constraints: the discreteness of $\boldsymbol{Y}$ induced by the quantization channel ϕ_q, and the discreteness of $\boldsymbol{U}$ and $\boldsymbol{V}$ imposed by their respective priors $p_{\mathbf{U}}(\boldsymbol{U})$ and $p_{\mathbf{V}}(\boldsymbol{V})$. In this thesis, we build upon the approximate message passing (AMP) paradigm and introduce a new related algorithm along with its state evolution analysis to solve the two types of discrete constraints involved in the MC problem.

The rest of this thesis is organized as follows: Chapter 1 reviews some of the very well-known latest research studies on discrete MF/MC problems, pointing out their limitations, highlighting the contribution of this research, and showcasing the novel approach used to surpass existing techniques. Then, Chapter 2 introduces the probabilistic graphical modeling technique, and gives some mathematical preliminaries that were used in the report. Chapter 3 delves into the AMP algorithms for linear compressed sensing applications. After that, Chapter 4 presents our proposed method for the quantized MF problem as also used to solve the discrete MC problem. Finally, Chapter 5 discusses simulation results of synthetic and real-world discrete datasets, demonstrating that our proposed method outperforms current state-of-the-art techniques.

Chapter 1

Related Work & Contributions

1.1 Related work & limitations

Until recent years, MC solutions (e.g., see [17,18] and references therein) focused on the reconstruction of real-valued matrices. Recent research studies, however, have shown in some applications, that the reconstruction performance can be improved by leveraging the discreteness (e.g., binary or counts) of the observations as encountered in many real-world applications [19–21]. Given the voluminous amount of research on real-valued MC methods, we review here the notable approaches to solve discrete-valued MC problems, i.e., when the entries of $\boldsymbol{Y}$ belong to a finite discrete alphabet. In the sequel, we call the MC methods accounting for the discrete nature of $\boldsymbol{Y}$, $\boldsymbol{U}$ or $\boldsymbol{V}$ as "discreteness-aware" approaches, and those ignoring it as "discreteness-blind" approaches.

1.1.1 Discreteness-blind methods

The observed matrix $\boldsymbol{Y}$ in (4) is sparse because it usually has a large unknown set of entries Ω due to the selection channel $\phi_s(\cdot)$. This sparsity constraint, along with the cardinality variation of Ω makes the computation of distances between row and/or column vectors of $\boldsymbol{Y}$ biased [27], [28], [26]. As a result, it becomes difficult to draw definitive conclusions on the missing entries based on the row/column correlations in $\boldsymbol{Y}$. In the context of movie recommendation systems [26], the zero-elements were replaced with the users' and items' average rating scores before applying the SVD to reduce the dimensionality of $\boldsymbol{Y}$. Another line of work in [27] has proposed an MF model, called LLORMA, based on SVD, assuming that $\boldsymbol{Y}$ is locally low-rank and hence modeled as a weighted sum of low-rank matrices. Meanwhile, Goldberg et al. in [28] addressed sparseness by ensuring that all users

rate a common set of k-items from Jester, an online joke-recommending system. Their method uses a principal component analysis (PCA) approach which reduces the dimensionality of $\boldsymbol{Y}$ by optimally projecting highly correlated data along a smaller number of orthogonal dimensions.

The probabilistic MF (PMF) model was also proposed in [29] and demonstrated better accuracy than other approaches, such as nearest-neighbor models and restricted Boltzmann machines. The efficiency of training PMF models derives from obtaining only point estimates of model parameters and hyper-parameters rather than inferring the entire posterior distribution over them.

While traditional SVD and PCA methods ignore the possible non-linearity inherent in data, nonlinear dimensionality reduction methods have been proposed such as sparse spectral techniques (e.g., locally linear embedding (LLE) [30] and Laplacian eigenmaps (LE) [31]). However, their formulations as eigenproblems have high condition numbers, making the attempt to solve them, using state-of-the-art eigensolvers, challenging. Traditional MF methods such as NMF [24, 32] handle the data sparsity by preprocessing the sparse observed matrix $\boldsymbol{Y}$. To obtain a dense matrix $\boldsymbol{Y}$, they impute appropriate values for missing entries before estimating a suitable low-rank approximation to $\boldsymbol{Y}$. While such an approach improves the reconstruction performance over the standard SVD method, the data sparsity is handled in a way that is not part of a well-studied mathematical framework where the processing operations are theoretically justified.

1.1.2 Discreteness-aware methods

Recently, there has been a lot of attention on finding ways to solve equation (1), when the observation matrix $\boldsymbol{Y}$ is discrete. The most efficient method for doing so is through bilinear factorization [34,35], which can directly deal with the discrete optimization problem and thus reduce the quantization loss to a large extent. Despite this achievement, discrete MF (DMF) is still limited by the distorted modeling of the real-world data structure. Firstly, DMF mainly focuses on modeling the observed entries but ignores the gathering structures that are hidden in the row/column features. Several research studies focused on the one-bit quantization scenario of the observation model (1), i.e., with $\phi_q(\cdot) = \text{sign}(\cdot)$. In [56], a solution for one-bit MC was proposed using convex programming that maximizes a log-likelihood function. Another one-bit MC solution was proposed in [57] where the authors examined max-norm convex relaxation, introduced a

max-norm constrained maximum likelihood estimate, and analyzed its convergence rate. Using information-theoretical methods, they established a mini-max lower bound, which resulted in an optimal Frobenius norm loss rate. The study in [58] presented a rank-constrained maximum likelihood estimator and an efficient greedy algorithm extending the conditional gradient descent for one-bit MC. The algorithm converges linearly to the underlying low-rank matrix, achieving the optimal statistical estimation error rate. Thus, while suffering from less generality in the quantization levels, this method is not robust in dealing with generalized quantized settings due to the priors on the latent factor matrices.

More recently, there have been other studies focused on extending one-bit MC to the multi-level quantized MC, such as [59]. There, a new method was introduced for quantized MC by relying on the approximate projected gradient approach to minimize a log-likelihood function under an exact rank constraint. To do this, the log-likelihood term is penalized with a log-barrier function. Bilinear factorization is then applied using the gradient descent technique to solve the resulting unconstrained problem. This method is robust and leads to noticeable numerical results in matrix recovery. However, it is vulnerable to convergence issues with local minima or saddle points and requires an accurate rank estimation at the beginning. In a more recent study [60], a novel technique for quantized MC was introduced. It employs a regularized convex cost function that combines a log-likelihood term and a trace norm term, using the bilinear factorization approach and the augmented Lagrangian method (ALM) to find the global minimizer and recover the original data.

Newly, inspired by proximal gradient techniques used in non-convex problems, Hiroki et al. [21] showed a good performance on the MovieLens dataset, They utilized an additive discrete-aware low-rank regularizer to handle the partiality of the observed data. To ensure that the real-valued entries of the matrix $\boldsymbol{U}\,\boldsymbol{V}^\top$ match with the discrete entries of $\boldsymbol{Y}$, the low-rank regularizer approach employs a mask operator (i.e., projection). By relaxing the original rank minimization problem, a convex optimization problem is obtained, which minimizes the sum of the singular values of the unknown matrix over the ℓ_1-norm using semi-definite programming. Another significantly more general line of work was recently proposed in [19] to handle both count and binary matrices in a unified manner via Poisson/truncated Poisson factor models and allows incorporating arbitrary types of row/column features via a regression model while maintaining robustness. An-

other proposal by Huo, Z et al. [20] presents a new optimal discrete-aware MC algorithm. They introduce new threshold variables that integrate continuous MC and threshold learning in the same loss function. Their work shows a low error estimation loss for discrete MC, enabling optimal threshold learning. Recently, in [61], the authors proposed a solution to the MC problem over finite fields of any arbitrary size. Their proposed technique is based on belief propagation (BP) and message passing (MP). While their solution is effective, AMP algorithms are computationally less expensive than MP algorithms on complete graphs. Additionally, AMP algorithms can also be used when more sophisticated priors (e.g., plug-and-play priors [62]) are available on the latent factors.

While most of the novel discreteness-aware methods have shown improved performance, they only treat the observation data matrix as discrete, but in some cases, the discreetness can be extended to the factorized matrices for even better results. To overcome all the aforementioned limitations, this work introduces a new AMP-like algorithm inspired by the BiG-VAMP introduced in [1], along with its SE analysis to solve the MC problem when $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{Y}$ are discrete.

1.2 Contributions

We build upon the bilinear generalized vector approximate message passing (BiG-VAMP) framework [1] to devise a new algorithm coined BiG-VAMP-MC tailored to solve discrete MC problems, along with its state evolution analysis. While existing MC methods handle discrete-valued entries either in the observation matrix $\boldsymbol{Y}$ or latent factors $\boldsymbol{U}$ and $\boldsymbol{V}$, BiG-VAMP-MC handles both cases simultaneously by:

1. integrating an output quantization and selection transformations, $\phi_q(\cdot)$ and $\phi_s(\cdot)$, respectively to model discrete entries and handle missing observations in $\boldsymbol{Y}$.

2. using adaptive quantization thresholds of $\phi_q(\cdot)$ learned via the EM algorithm.

3. taking advantage of the discrete-valued i.i.d. priors on $\boldsymbol{U}$ and $\boldsymbol{V}$ which are systematically handled by BiG-VAMP[1].

[1]Note that a state-of-the-art competitor to BiG-VAMP, namely BiG-AMP [37], does not converge under discrete priors as has been shown in [1].

Chapter 2

Graphical models

Introduction

In this chapter, we start by introducing some fundamentals and mathematical tools that were used along with this research project, from graph theory to belief propagation, and all the parts that are detailed in later sections.

2.1 Graphical modeling

Probabilistic graphical modeling is a part of artificial intelligence that reviews how to utilize probability distributions to describe the world and to make helpful predictions about it, there are many motivations to find out about probabilistic modeling. For one, it is an interesting scientific field with an excellent theory that spans in surprising ways two different branches of mathematics probability and graph theory. Probabilistic modeling additionally has interesting associations with philosophy, especially the topic of causality. At the same time, probabilistic modeling is broadly utilized through machine learning and in many real-world applications. These strategies can be used to solve problems in fields as diverse as medicine, language processing, vision, and many others. This mix of elegant theory and ground-breaking applications makes graphical models one of the most intriguing themes in modern artificial intelligence and software engineering

When trying to solve a real-world problem using mathematics, it is exceptionally regular to characterize it with a mathematical model of the world in the form of an equation. Perhaps the simplest model would be a linear equation of the form:

$$\mathbf{y} = \boldsymbol{\beta}^{\top}\mathbf{x}. \tag{2.1}$$

where $\mathbf{y}$ is the variable that we want to predict, and $\mathbf{x}$ are known (given) variables that affect the outcome. For example, $\mathbf{y}$ may be the compressed signal observed via certain antennas, and $\mathbf{x}$ is the true signal that we want to recover, etc. Here $\mathbf{y}$ is a linear function of this input (parameterized by $\boldsymbol{\beta}$).

Often, the real world that we want to model is very complicated, in particular, it regularly represents a significant amount of uncertainty. It is in this way very natural to deal with this problem by modeling the world as a probability distribution $p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$ that represents the joint probability of the variables, which follow a known distribution for the most of problems. To go deeper through the details of this theory next subsections are divided into two major parts: representation (how to specify a model) and inference (how to ask the model questions).

2.2 Describing probabilities with graphs

A perfect way to represent the mathematic model that represents the real-world problem is to use the graph theory; probabilities are described by graphs whose properties (e.g., connectivity, tree-width) will reveal probabilistic and algorithmic features of the model (e.g., independence, complexity), in which the nodes correspond to variables and the edges indicate dependency relationships.

2.2.1 Probabilistic modeling with Bayesian networks

The sorts of models that we see here are alluded to as Bayesian networks. Bayesian networks (a.k.a. Directed graphical models) are a group of probability distributions that admit a compact parametrization that can be normally described using a directed graph. The main idea behind this parametrization is simple, recall that by the chain rule, we can write any probability p as:

$$p_{\mathbf{x}_1,\cdots,\mathbf{x}_n}(x_1, x_2, \ldots, x_n) = p_{\mathbf{x}_1}(x_1)\, p_{\mathbf{x}_2|\mathbf{x}_2}(x_2|x_1) \cdots p_{\mathbf{x}_n|\mathbf{x}_{n-1},\cdots,\mathbf{x}_1}(x_n|x_{n-1}, \ldots, x_2, x_1)\,. \tag{2.2}$$

A compact Bayesian network is a distribution in which each factor on the right-hand side depends only on a small number of ancestor variables $\mathbf{x}_{A_i}$:

$$p_{\mathbf{x}_i|\mathbf{x}_{i-1},\cdots,\mathbf{x}_1}(x_i|x_{i-1}, \ldots, x_1) = p_{\mathbf{x}_i|\mathbf{x}_{A_i}}(x_i|x_{A_i})\,. \tag{2.3}$$

More formally, a Bayesian network is a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ whose nodes represent variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not associated (no path interfaces one node to another) represent variables that are conditionally independent of each other. Each node is related to a probability function that takes, as input, a particular set of values for the node's parent variables and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node. Figure 2.1 shows a Bayesian network over four variables

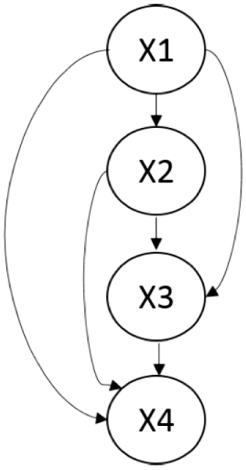


Figure 2.1: A fully connected Bayesian network over four variables

with :

- A random variable $\mathbf{x}_i$ for each node $i \in \mathbf{V}$.
- One conditional probability distribution $p_{\mathbf{x}_i|\mathbf{x}_{A_i}}(x_i|x_{A_i})$ per node, specifying the probability of $\mathbf{x}_i$ conditioned on its parents values.

To sum up, Bayesian networks represent probability distributions that can be formed via products of smaller, local conditional probability distributions (one for each variable). By communicating a probability in this form, we are introducing into our model assumptions that certain variables are independent.

This brings up the issue: which independence suppositions are we precisely making by using a Bayesian network model? This question is important because we should know definitely what model suppositions we are making (and whether they are correct). Additionally, this information will help us design more proficient inference algorithms later on.

2.2.2 Factor graphs

Bayesian networks are a class of models that can moderately represent many interesting probability distributions. However, some distributions may have independence suppositions that cannot be perfectly represented by the structure of a Bayesian network.

In such cases, unless we want to present fake independencies among the variables of our model, we must fall back to a less reduced representation (which can be viewed as a graph with extra, superfluous edges). This leads to additional, unnecessary parameters in the model, and makes it progressively hard to get familiar with and to make predictions.

There exists, however, another method for compactly representing probability distribution which is based on undirected graphs, a factor graph is one such approach to this. A factor graph is a bipartite graph where one group is the variables in the distribution being displayed, and the other group is the factors defined by these variables. Edges go among factors and variables that those factors depend on.

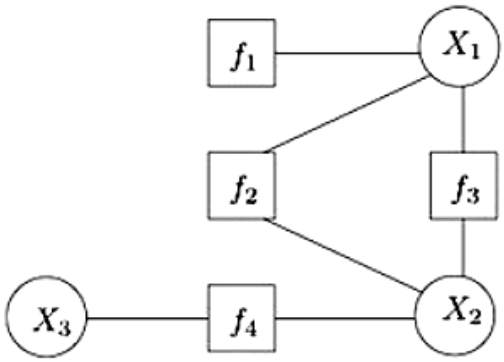


Figure 2.2: Example of a factor graph with three variables and four factors

The graph displayed in Fig. 2.2 represents the following joint probability distribution:

$$p_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3}(x_1, x_2, x_3) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_1, x_2)\, f_4(x_2, x_3)\,. \tag{2.4}$$

This view permits us to more readily observe the factor dependencies between variables, and later we will see it allows us to process some probability distributions

without any problem.

2.3 Inference in graphical models

Given a probabilistic model, the inference points out a way to obtain answers to relevant questions about the real modeled world. Such questions frequently reduce to querying the marginal or conditional probabilities of certain events of interest. More concretely, it will be typically interested in asking the system two types of questions:

- Marginal inference: how much is the probability of a given variable in the model after summing everything else out?

$$p_{\mathbf{x}_1}(x_1) = \sum_{x_2} \sum_{x_3} \cdots \sum_{x_n} p_{\mathbf{x}_1,\cdots,\mathbf{x}_n}(x_1, x_2, \ldots, x_n). \tag{2.5}$$

- Maximum a posteriori (MAP) inference asks for the most likely assignment of variables.

$$\underset{x_1,\ldots,x_n}{\operatorname{argmax}}\, p_{\mathbf{x}_1,\cdots,\mathbf{x}_n,\mathbf{y}}(x_1, \ldots, x_n, \mathbf{y} = 1). \tag{2.6}$$

The main difference between the two types of inference on graphs cited above is that the marginal inference treats the estimation problems and recovering certain distributions that model a real-life problem, and the MAP inference goes through optimization problems when it works on the maximization of the joint probability of specific phenomena under some existing constraints.

2.4 Belief propagation

Belief propagation (BP), also known as sum-product message passing, is an algorithm treating inference on graphical models, such as Bayesian networks and Factor graphs. It computes the marginal distribution for each unobserved variable, conditional on any observed variables.

Variants of the BP algorithm exist for several types of graphs, we describe here the one used in our main algorithm which treats estimation problems and operates on a factor graph. The algorithm works by exchanging messages (real-valued functions) between the hidden nodes alongside the edges. More precisely, these messages contain the "impact" that one variable applies to another. The

messages are computed distinctively relying upon whether the node getting the message is a variable node or a factor node, we have two types of messages: factor-to-variable messages μ and variable-to-factor messages ν. Both messages require taking a product, but only the factor-to-variable messages μ require summing, the calculation of these messages is given by the following formulas:

$$\begin{aligned} \nu_{var(i)\to fac(s)}\left(x_i\right) &= \prod_{t\in N(i)\backslash s} \mu_{fac(t)\to var(i)}\left(x_i\right), \\ \mu_{fac(s)\to var(i)}\left(x_i\right) &= \sum_{x_{N(s)|i}} f_s\left(x_{N(s)}\right) \prod_{j\in N(s)\backslash i} \nu_{var(j)\to fac(s)}\left(x_j\right). \end{aligned} \tag{2.7}$$

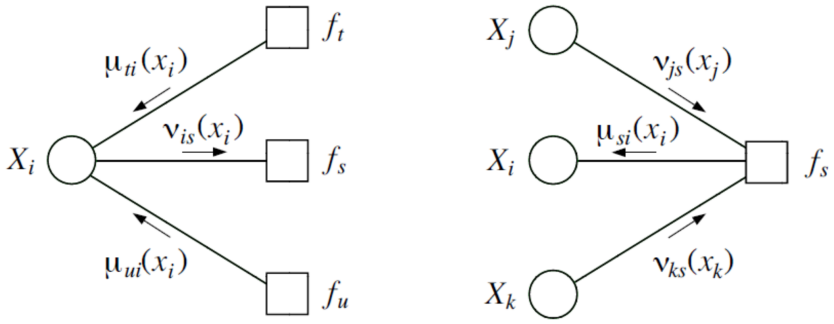


Figure 2.3: Factor graph messages

The notation $N(i)\backslash j$ refers to the set of nodes that are neighbors of i, excluding j.

If convergence happens, the estimated marginal distribution of each node $\mathbf{x}_i$ is proportional to the product of all messages from adjoining factors (where $\propto$ denotes the proportional operator as defined in Notation part)

$$p_{\mathbf{x}_i}\left(\mathrm{x}_i\right) \propto \prod_{t\in N(i)} \mu_{fac(t)\to var(i)}\left(\mathrm{x}_i\right). \tag{2.8}$$

Similarly, the estimated joint marginal distribution of the set of variables $\mathbf{x}_{N(s)}$ related to one factor f_s is proportional to the product of the factor and the messages from the variables

$$p_{\mathbf{x}_{N(s)}}\left(\mathbf{x}_{N(s)}\right) \propto f_s\left(\mathbf{x}_{N(s)}\right) \prod_{j\in N(s)} \nu_{var(j)\to fac(s)}\left(\mathrm{x}_j\right). \tag{2.9}$$

In the case of factor tree graphs, these estimated marginal distributions converge to the true distributions in limited iterations.

2.5 Mathematical preliminaries

Pseudo-Lipschitz functions

For a given $k \geq 1$, a function $\mathrm{g} : \mathbb{R}^s \to \mathbb{R}^r$ is called pseudo-Lipschitz of order k, if there exists a positive constant $C > 0$ such that for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^s$

$$\|\mathbf{g}(\mathbf{x}_1) - \mathbf{g}(\mathbf{x}_2)\| \leq C \|\mathbf{x}_1 - \mathbf{x}_2\| \left[1 + \|\mathbf{x}_1\|^{k-1} + \|\mathbf{x}_2\|^{k-1}\right]. \tag{2.10}$$

In the case where $k = 1$, the pseudo-Lipschitz continuity becomes the standard known Lipschitz continuity.

Empirical convergence

For a given dimension $r > 0$, and suppose that, for each $N, \mathrm{y}(N)$ is a vector of the form

$$\mathbf{y}(N) = [\mathbf{y}_1(N), \ldots, \mathbf{y}_N(N)]$$

with vector sub-components $\mathbf{y}_n(N) \in \mathbb{R}^r$. Thus, the dimension of $\mathrm{y}(N)$ is $r \times N$. In this case, $\mathrm{y}(N)$ is called a block vector sequence. Now, we suppose that $\mathbf{y} = \mathbf{y}(N)$ is a block vector sequence that may be deterministic or random. For a fixed $p \geq 1$, We say $\mathbf{y} = \mathbf{y}(N)$ converges empirically with p-th order moments if there exists a random variable $Y \in \mathbb{R}^r$ such that

(i) $\mathbb{E}|Y|^p < \infty$; and
(ii) for any scalar-valued pseudo-Lipschitz continuous function $g(\cdot)$ of order p

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} g(\mathbf{y}_n(N)) = \mathbb{E}[g(Y)] \text{ almost surely .}$$

Thus, the empirical mean of the components $g(\mathbf{y}_n(N))$ converges to the expectation $\mathbb{E}[g(Y)]$. When y converges empirically with p-th order moments, it will be written, with some abuse of notation,

$$\lim_{N\to\infty} \{\mathbf{y}_n\}_{n=1}^{N} \overset{PL(p)}{=} Y.$$

Central limit theorem (CLT)

Let $(Y_n)_{n\geq 1}$ be a sequence of real random variables i.i.d. of integrable square of the same distribution as Y. We assume that $\mathrm{Var}(Y) > 0$ and we let

$$\hat{m}_N := \frac{1}{N}\sum_{n=1}^{N} Y_n,\ \hat{\sigma}_N := \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(Y_n - \hat{m}_N)^2},$$

thus,

$$\sqrt{N}\left(\frac{\hat{m}_N - E[Y]}{\hat{\sigma}_N}\right) \longrightarrow \mathcal{N}(0,1) \text{ in law,}$$
$$\text{when } N \longrightarrow \infty.$$

In particular let a real $c > 0$

$$P\left[\sqrt{N}\left|\frac{\hat{m}_N - E[Y]}{\hat{\sigma}_N}\right| < c\right] \longrightarrow 1 - \alpha_c,$$

$$\text{when } N \longrightarrow \infty,$$

where $\alpha_c = P[|X| < c]$ for $X \sim \mathcal{N}(0,1)$ so for N large,

$$P\left[E[Y] \in \left[\hat{m}_N \pm c\frac{\hat{\sigma}_N}{\sqrt{N}}\right]\right] = 1 - \alpha_c.$$

Asymptotic theorem on random matrix theory

In the book "Random Matrix Theory and Wireless Communications" [6], the author proposed a tutorial overview that focuses on linear vector memoryless channels. These channels have the form $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{H} \in \mathbb{R}^{N\times K}$, $\boldsymbol{x} \in \mathbb{R}^K$, $\boldsymbol{y} \in \mathbb{R}^N$, and $\boldsymbol{n} \in \mathbb{R}^N$ is a circularly symmetric Gaussian noise. The theorem specifically considers the case of a zero-mean, independent, and identically distributed (i.i.d.) channel matrix $\boldsymbol{H}$ with variance $\frac{1}{N}$ in the high-dimensional regime, where $K, N \to \infty$ with $\frac{K}{N} \to \beta$. For a given signal-to-noise ratio (SNR), the tutorial shows that the following result holds:

$$\frac{1}{K}\,\mathrm{tr}\left\{\left(\mathbf{I} + \mathrm{SNR}\,\mathbf{H}^\dagger\mathbf{H}\right)^{-1}\right\} \to 1 - \frac{\mathcal{F}(\mathrm{SNR},\beta)}{4\beta\,\mathrm{SNR}}, \tag{2.11}$$

with:

$$\mathcal{F}(x,z) = \left(\sqrt{x(1+\sqrt{z})^2+1} - \sqrt{x(1-\sqrt{z})^2+1}\right)^2.$$

Conclusion

In this chapter, we provided the fundamental tools of graphical models, and some necessary mathematical preliminaries and background, that we used in this work, especially, in the derivation of our proposed computational and theoretical solutions.

Chapter 3

Linear Approximate Message Passing Algorithms

Introduction

In this chapter, we present a quick overview of compressed sensing applications in the signal processing field. Furthermore, we briefly provide detailed techniques for signal recovery using message passing algorithms, applied to linear vector memoryless channel models.

3.1 Compressed sensing

Compressed sensing gives provides a framework for recuperating signals from compressed measurements. For example, it can exploit the structure of natural pictures and recover a picture from only a few random measurements. the received measurements can be expressed linearly or non-linearly as a function of the signal to be recovered with additive noise most of the time. For instance, in signal processing, it treats the linear inverse problem models of the form:

$$\mathbf{y} = \mathbf{A}\mathbf{x}_0 + \mathbf{w}, \tag{3.1}$$

where $\mathbf{y}$ is M-dimensional measurements that contain the received compressed signal entries, $\mathbf{x}_0$ is the N-dimensional true signal to recover, the M-dimensional vector $\mathbf{w}$ represents the additive circularly symmetric Gaussian noise and the matrix $\mathbf{A}$ is the $M \times N$ random channel matrix used for compression. To solve this problem, it is generally assumed that the sensing matrix $\mathbf{A}$ is fat (More unknowns than measurements $N >> M$) and that the true signal $\mathbf{x}_0$ to recover

has a sparse structure.

3.2 Approximate message passing

There has been as of late much exertion on finding new algorithms for recuperating sparse solutions from the measurements of the form (3.1), the AMP algorithm was proposed as an iterative method to recover $\mathbf{x}_0 \in \mathbb{R}^N$ based on BP inference technique, from the received values of $\mathbf{y} \in \mathbb{R}^M$, when $\mathbf{A} \in \mathbb{R}^{M\times N}$ is a large i.i.d. sub-Gaussian matrix with variance $\frac{1}{M}$, and the noise vector $\mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, \gamma_{w0}^{-1}\mathbf{I}\right) \in \mathbb{R}^M$. One approach to do that is by solving an optimization problem of the form:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{3.2}$$

where $\hat{\mathbf{x}}$ is an estimate of $\mathbf{x}_0$, with $\lambda > 0$ promotes the sparsity in $\hat{\mathbf{x}}$, in order to introduce BP, we need to define a joint density distribution on the variables $s = (s_1, \ldots, s_N)$ as follows:

$$\mu(\mathbf{d}s) = \frac{1}{C}\prod_{i=1}^{N} \exp\left(-\beta\lambda\left|s_i\right|\right) \prod_{a=1}^{M} \exp\left\{-\frac{\beta}{2}\left(y_a - (As)_a\right)^2\right\} \mathbf{d}s, \tag{3.3}$$

where C is the normalization constant, and β is a positive parameter when it goes to ∞, the mass of this distribution converges for the solution of equation (3.1). Here, we consider the factor graph $G = (V, F, E)$ with variable nodes $V = [N] \equiv \{1, 2, \ldots, N\}$, factor nodes $F = [M] \equiv \{1, 2, \ldots, M\}$ and edges $E = [N] \times [n] = \{(i, a) : i \in [N], a \in [M]\}$. Thusly G is a factor graph with N variable nodes and M functional nodes. It is simple to deduce that the joint distribution (3.3) is defined according to this factor graph. So, the messages exchanged between nodes and factors in the sum-product algorithm will have the following expressions:

$$\begin{aligned} \nu_{i\to a}^{t+1}\left(s_i\right) &\cong \exp\left(-\beta\lambda\left|s_i\right|\right) \prod_{b\neq a} \nu_{b\to i}^{t}\left(s_i\right), \\ \hat{\nu}_{a\to i}^{t}\left(s_i\right) &\cong \int \exp\left\{-\frac{\beta}{2}\left(y_a - (As)_a\right)^2\right\} \prod_{j\neq i} \mathbf{d}\nu_{j\to a}^{t}\left(s_j\right). \end{aligned} \tag{3.4}$$

Then, we can show, by applying the CLT, that in the large system limit when $N, M \to \infty$ with $\frac{M}{N} = \delta$, the BP messages are well approximated by families with two scalar parameters, after deriving the updated rules for these parameters and taking $\beta \to \infty$ (the entire mass of the distribution will concentrate around

the mode) to get the appropriate rules for minimization. Finally, Taylor formulas approximate the message passing rules for large systems with updates. We will get the iterative algorithm shown below:

$$x_t = \eta\left(x_t + A^* z_t; \lambda + \gamma_t\right), \tag{3.5}$$

$$z_{t+1} = y - Ax_t + \frac{1}{\delta} z_t \left\langle \eta'\left(x^{t-1} + A^* z^{t-1}\right)\right\rangle, \tag{3.6}$$

$$\gamma_{t+1} = \frac{\lambda + \gamma_t}{\delta} \left\langle \eta'\left(Az_t + x_t; \gamma_t + \lambda\right)\right\rangle, \tag{3.7}$$

with $\eta(,)$ is the soft thresh-holding function defined as follows:

$$\begin{aligned} \eta(x; b) &= \arg\min_{s\in\mathbb{R}} \left\{ |s| + \frac{1}{2b}(s-x)^2 \right\}, \\ &= \begin{cases} x - b & \text{if } b < x \\ 0 & \text{if } -b \le x \le b \\ x + b & \text{if } x < -b \end{cases} \end{aligned} \tag{3.8}$$

where $\langle\cdot\rangle$ is the empirical average operation as defined in the notation part.

Another approach to do that is by using the Bayesian inference. In this case, one takes a prior density $p_{\mathbf{x}}(\mathbf{x})$ and likelihood function $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$ and then tries to determine the posterior density:

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \frac{p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) p_{\mathbf{x}}(\mathbf{x})}{\int p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) \mathbf{dx}}. \tag{3.9}$$

or, in practice, the MAP estimate:

$$\hat{\mathbf{x}}_{\mathbf{MAP}} = \arg\max_{\mathbf{x}} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}). \tag{3.10}$$

or the minimum mean-squared error (MMSE) estimate:

$$\hat{\mathbf{x}}_{\mathbf{MMSE}} = \arg\min_{\widetilde{\mathbf{x}}} \int \|\mathbf{x} - \widetilde{\mathbf{x}}\|^2 p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \mathbf{dx} = \mathbb{E}_{\mathbf{x}|\mathbf{y}}\{\mathbf{x}|\mathbf{y}\}. \tag{3.11}$$

using the posterior marginal densities $\left\{p_{\mathbf{x_n}|\mathbf{y}}\left(x_n|\mathbf{y}\right)\right\}_{n=1}^{N}$.

For this approach, we need a function called denoising function $g_1\left(\cdot, \gamma_t\right) : \mathbb{R}^N \to \mathbb{R}^N$ parameterized by γ_t, and $\left\langle \mathbf{g}_1'\left(\mathbf{r}_t, \gamma_t\right)\right\rangle$ is its divergence at $\mathbf{r}_t$. In par-

ticular, $\mathbf{g}_1'(\mathbf{r}_t, \gamma_t) \in \mathbb{R}^N$ is the diagonal of the Jacobian:

$$\mathbf{g}_1'(\mathbf{r}_t, \gamma_t) = \operatorname{diag}\left[\frac{\partial \mathbf{g}_1(\mathbf{r}_t, \gamma_t)}{\partial \mathbf{r}_t}\right]. \tag{3.12}$$

when $\mathbf{A}$ is a large i.i.d. sub-Gaussian matrix, $w \sim \mathcal{N}\left(\mathbf{0}, \gamma_{w0}^{-1}\mathbf{I}\right)$, and $\mathbf{g}_1(\cdot, \gamma_t)$ is separable , i.e.:

$$[\mathbf{g}_1(\mathbf{r}_t, \gamma_t)]_n = g_1(r_{tn}, \gamma_t); \forall n, \tag{3.13}$$

with identical Lipschitz components $g_1(\cdot, \gamma_t) : \mathbb{R} \to \mathbb{R}$, AMP shows a momentous conduct, which is that $\mathbf{r}_t$ behaves like a white-Gaussian-noise corrupted version of the true signal $\mathbf{x}_0$:

$$\mathbf{r}_t = \mathbf{x}_0 + \mathcal{N}(\mathbf{0}, \tau_t \mathbf{I}), \tag{3.14}$$

for some variance $\tau_t > 0$, the expression of the denoising function is deduced recording to the problem treated, supposing that $\mathbf{x}$ has an i.i.d. prior, which means:

$$p_{\mathbf{x}}(\mathbf{x}) = \prod_{n=1}^{N} p_{\mathbf{x_n}}(x_n). \tag{3.15}$$

For the MMSE problem cited above, AMP can be applied by choosing:

$$g_1(r_{tn}, \gamma_t) = \mathbb{E}[x_n | r_{tn}, \gamma_t], \tag{3.16}$$

where the conditional density is given by:

$$p_{\mathbf{x_n}|\mathbf{t_{tn}}}(x_n | r_{tn}, \gamma_t) \propto \exp\left[-\frac{\gamma_t}{2}|r_{tn} - x_n|^2 + \log\left(p_{\mathbf{x_n}}(x_n)\right)\right]. \tag{3.17}$$

For later use in the AMP algorithm we note that the derivative of the MMSE scalar denoiser (3.16) is calculated through the following expression:

$$g_1'(r_{tn}, \gamma_t) = \gamma_t \operatorname{var}[x_n | r_{tn}, \gamma_t]. \tag{3.18}$$

In (3.17) γ_t can be interpreted as an estimate of τ_t^{-1} the iteration- t precision of $\mathbf{r}_t$ from (3.14). In the case that τ_t is known, the "matched" condition is satisfied when $\gamma_t = \tau_t^{-1}$.

The AMP iterative method as shown in Algorithm 1, has a great convergence performance for large Gaussian i.i.d. channel matrices and it can be analyzed theoretically via state evolution framework which is introduced in the next parts, However, in cases where the channel matrix is not just Gaussian i.i.d., the algo-

rithm diverges and might lose some key properties, so for a better way there is an alternative algorithm that was recently proposed, which is defined in the later section.

Algorithm 1 AMP

Require : Matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, measurement vector $\mathbf{y}$, denoiser $\mathbf{g}_1(\cdot, \gamma_t)$, and number of iterations T_{max}.

Set $v_{-1} = 0$ and select initial $\mathbf{r}_0, \gamma_0$
for $t = 1, \ldots, T_{max}$ **do**

$\hat{\mathbf{x}}_t = \mathbf{g}_1(\mathbf{r}_t, \gamma_t)$
$\alpha_t = \langle \mathbf{g}_1'(\mathbf{r}_t, \gamma_t) \rangle$
$\mathbf{v}_t = \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_t + \frac{N}{M}\alpha_{t-1}\mathbf{v}_{t-1}$
$\mathbf{r}_{t+1} = \hat{\mathbf{x}}_t + \mathbf{A}^\top \mathbf{v}_t$
Select γ_{t+1}

end for
Return $\hat{\mathbf{x}}_{T_{max}}$

3.3 Vector approximate message passing

The vector AMP algorithm (VAMP) was proposed as a better alternative to AMP, which works perfectly and guarantees all the desirable properties of the original AMP (i.e., low complexity, very fast convergence, ...), but for a much wider general class of channel matrices $\mathbf{A}$, those are called right orthogonally invariant matrices: when the distribution of $\mathbf{A}$ is the same as $\mathbf{AV}$ for any fixed orthogonal matrix $\mathbf{V}$.

The main idea of its derivation is to split the variable $\mathbf{x}$ into two identical variables $\mathbf{x}_1$ and $\mathbf{x}_2$, the first variable $\mathbf{x}_1$ will be denoised by the MMSE bloc using the function defined in (3.16), (3.18), and the second variable $\mathbf{x}_2$ will be estimated using the LMMSE (Linear-MMSE) estimator bloc, after exchanging the approximated messages with the MMSE bloc. Figure 3.1 shows the new factor graph used in the VAMP construction, where its new joint distribution is given by:

$$p_{\mathbf{y},\mathbf{x_1},\mathbf{x_2}}(\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2) = p_{\mathbf{x_1}}(\mathbf{x}_1)\, \delta(\mathbf{x}_1 - \mathbf{x}_2)\, \mathcal{N}\left(\mathbf{y}; \mathbf{A}\mathbf{x}_2, \gamma_w^{-1}\mathbf{I}\right), \tag{3.19}$$

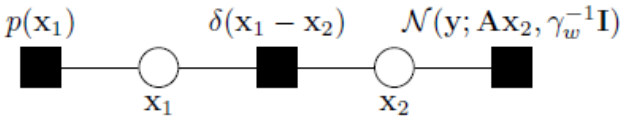


Figure 3.1: Vector approximate message passing factor graph

where $\delta(\cdot)$ is the Dirac delta distribution. Using the BP techniques for the two sides of the graph, and by applying the above message passing rules to the new factor graph, we will get the VAMP algorithmic steps shown in Algorithm 2.

Here is the expression of the LMMSE denoising function:

Algorithm 2 VAMP

Require : LMMSE estimator $\mathbf{g}_2(\mathbf{r}_{2t}, \gamma_{2t})$ from (24), denoiser $\mathbf{g}_1(\cdot, \gamma_{1t})$, and number of iterations T_{max}.

Select initial $\mathbf{r}_{10}$ and $\gamma_{10} \geq 0$
for $t = 1, \ldots, T_{max}$ **do**

// Denoising
$\hat{\mathbf{x}}_{1t} = \mathbf{g}_1(\mathbf{r}_{1t}, \gamma_{1t})$
$\alpha_{1t} = \langle \mathbf{g}_1'(\mathbf{r}_{1t}, \gamma_{1t}) \rangle$
$\eta_{1t} = \gamma_{1t}/\alpha_{1t}$
$\gamma_{2t} = \eta_{1t} - \gamma_{1t}$
$\mathbf{r}_{2t} = (\eta_{1t}\hat{\mathbf{x}}_{1t} - \gamma_{1t}\mathbf{r}_{1t})/\gamma_{2t}$

// LMMSE estimation
$\hat{\mathbf{x}}_{2t} = \mathbf{g}_2(\mathbf{r}_{2t}, \gamma_{2t})$
$\alpha_{2t} = \langle \mathbf{g}_2'(\mathbf{r}_{2t}, \gamma_{2t}) \rangle$
$\eta_{2t} = \gamma_{2t}/\alpha_{2t}$
$\gamma_{1,t+1} = \eta_{2t} - \gamma_{2t}$
$\mathbf{r}_{1,t+1} = (\eta_{2t}\hat{\mathbf{x}}_{2t} - \gamma_{2t}\mathbf{r}_{2t})/\gamma_{1,t+1}$

end for
Return $\hat{\mathbf{x}}_{1T_{max}}$

$$\mathbf{g}_2(\mathrm{r}_{2t}, \gamma_{2t}) := \left(\gamma_w \mathbf{A}^\top \mathbf{A} + \gamma_{2t}\mathbf{I}\right)^{-1} \left(\gamma_w \mathbf{A}^\top \mathbf{y} + \gamma_{2t}\mathbf{r}_{2t}\right). \tag{3.20}$$

Analogously to the MMSE function (3.16), the above expression can be recognized as the MMSE estimate of a random vector x_2 under likelihood $\mathcal{N}(\mathrm{y}; \mathrm{A}\mathrm{x}_2, \gamma_w^{-1}\mathrm{I})$

and prior $\mathrm{x}_2 \sim \mathcal{N}\left(\mathrm{r}_{2t}, \gamma_{2t}^{-1}\mathrm{I}\right)$. since this estimate is linear in r_{2t}, and it is easy to deduce its derivative expression as:

$$\langle \mathbf{g}_2'\left(\mathbf{r}_{2t}, \gamma_{2t}\right)\rangle = \frac{\gamma_{2t}}{N}\,\mathrm{Tr}\left[\left(\gamma_w \mathbf{A}^\top \mathbf{A} + \gamma_{2t}\mathbf{I}\right)^{-1}\right]. \tag{3.21}$$

Conclusion

In this chapter, we presented an important application of signal processing in wireless communication, linear compressed sensing, and we showed two important algorithms that were recently developed to treat these linear problems efficiently, with a superior performance.

Chapter 4

Quantized BiG-VAMP for discrete MC

Introduction

In this chapter, we start by formulating the bilinear problem recovery, and then we take a look at the background of low-rank matrix reconstruction. To end this chapter, we present our novel BiG-VAMP-MC algorithm, as a bilinear signal recovery tool that solves the discrete MF/MC problems.

4.1 Bilinear generalized signal recovery problems

4.1.1 Problem formulation

We consider an observation matrix, $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$, obtained from the following generalized bilinear model:

$$p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y}|\boldsymbol{Z}) \;=\; \prod_{i=1}^{N}\prod_{j=1}^{M} p_{\mathsf{y}_{ij}|\mathsf{z}_{ij}}(y_{ij}|z_{ij}) \quad \text{with} \quad \boldsymbol{Z} \;=\; \boldsymbol{U}\boldsymbol{V}^{\mathsf{T}}, \tag{4.1}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are two unknown matrices in $\mathbb{R}^{N \times r}$ and $\mathbb{R}^{M \times r}$, respectively. The goal is to recover $\boldsymbol{U}$ and $\boldsymbol{V}$ based on the knowledge of $\boldsymbol{Y}$ and the model in (4.1). The latter applies to a myriad of problems ranging from noisy dictionary learning [41], matrix completion [42], and sparse PCA [43], to matrix factorization [44], low-rank matrix reconstruction [36], and subgraph estimation [45], just to name a few. A special relevant case of the generalized bilinear observation model in (4.1)

is:

$$\boldsymbol{Y} \;=\; \phi\left(\boldsymbol{U}\,\boldsymbol{V}^\top + \boldsymbol{W}\right), \tag{4.2}$$

in which $\phi(\cdot)$ is a non-linear element-wise function and $\boldsymbol{W} \in \mathbb{R}^{N\times M}$ is an additive white Gaussian noise matrix whose entries are assumed to be mutually independent with mean zero and variance γ_w^{-1}, i.e., $w_{i,j} \;\sim \mathcal{N}(w_{ij}; 0, \gamma_w^{-1})$.

Consider the bilinear recovery of two random independent matrices $\mathsf{U} = [\mathsf{u}_1, \ldots, \mathsf{u}_N]^\top \in \mathbb{R}^{N\times r}$ and $\mathsf{V} = [\mathsf{v}_1, \ldots, \mathsf{v}_M]^\top \in \mathbb{R}^{M\times r}$ from a linear noisy observation $\boldsymbol{Y} = \boldsymbol{U}\,\boldsymbol{V}^\top + \boldsymbol{W} \in \mathbb{R}^{N\times M}$. Given some common priors, $p_{\mathsf{u}}(\boldsymbol{u}_i)$ and $p_{\mathsf{v}}(\boldsymbol{v}_i)$, on the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, respectively, the goal of BP is to approximate their joint posterior distribution:

$$\begin{aligned} p_{\mathsf{u},\mathsf{v}|\mathsf{Y}}(\boldsymbol{u}_i, \boldsymbol{v}_j|\boldsymbol{Y}; \gamma_w^{-1}, \beta) &= p_{\mathsf{u},\mathsf{v}|\mathsf{y}_{ij}}(\boldsymbol{u}_i, \boldsymbol{v}_j|y_{ij}; \gamma_w^{-1}, \beta) \\ &\propto \underbrace{p_{\mathsf{y}_{ij}|\mathsf{u},\mathsf{v}}\left(y_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j; \gamma_w^{-1}\right)^\beta}_{\triangleq\, f_{ij}(\boldsymbol{u}_i, \boldsymbol{v}_j)}\, p_{\mathsf{u}}(\boldsymbol{u}_i)^\beta\, p_{\mathsf{v}}(\boldsymbol{v}_j)^\beta. \end{aligned} \tag{4.3}$$

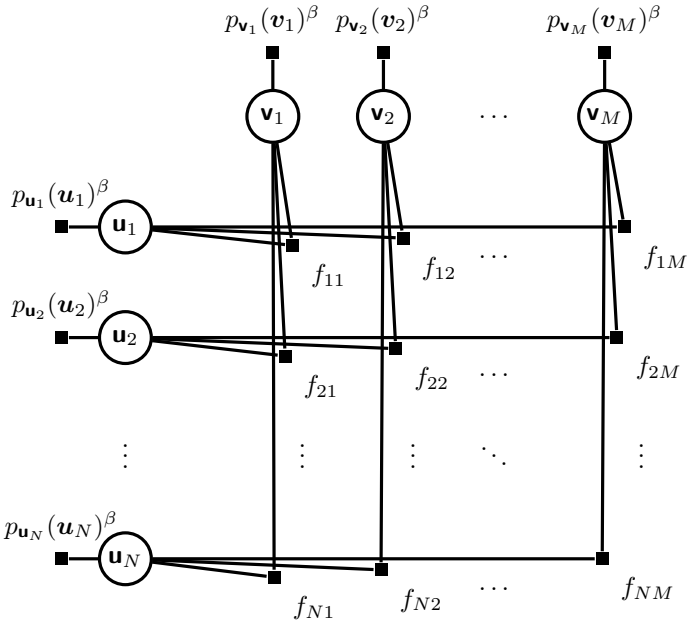
4.1.2 Low-rank matrix reconstruction: Graphical models and derivation

We consider the factor graph in Fig. 4.1 associated to (4.3) with variable nodes, u_i and v_j, their prior factor nodes, $p_{\mathsf{u}}(\boldsymbol{u}_i)^\beta$ and $p_{\mathsf{v}}(\boldsymbol{v}_j)^\beta$, and the labels, f_{ij}, which we use as a shorthand notations for the factor nodes:

$$\begin{aligned} f(\boldsymbol{u}_i, \boldsymbol{v}_j) &\triangleq p_{\mathsf{y}_{ij}|\mathsf{u},\mathsf{v}}\left(y_{ij}|\boldsymbol{u}_i, \boldsymbol{v}_j, \gamma_w^{-1}\right)^\beta, \\ &= \mathcal{N}\left(y_{ij}; \boldsymbol{u}_i^\top \boldsymbol{v}_j, \beta^{-1}\gamma_w^{-1}\right). \end{aligned} \tag{4.4}$$

wherein β is a temperature parameter introduced here to treat the MMSE ($\beta = 1$) and MAP ($\beta = +\infty$) inference problems in a unified framework. By taking $\beta = 1$ (i.e., MMSE estimation), $p_{\mathsf{u},\mathsf{v}|\mathsf{Y}}(\boldsymbol{u}_i, \boldsymbol{v}_j|\boldsymbol{Y}; \gamma_w^{-1}, \beta)$ reduces to the true joint posterior $p_{\mathsf{u},\mathsf{v}|\mathsf{Y}}(\boldsymbol{u}_i, \boldsymbol{v}_j|\boldsymbol{Y}; \gamma_w^{-1})$. In the limit $\beta \to \infty$ (i.e., MAP estimation), however, $p_{\mathsf{u},\mathsf{v}|\mathsf{Y}}(\boldsymbol{u}_i, \boldsymbol{v}_j|\boldsymbol{Y}; \gamma_w^{-1}, \beta)$ concentrates on the maxima of $p_{\mathsf{u},\mathsf{v}|\mathsf{Y}}(\boldsymbol{u}_i, \boldsymbol{v}_j|\boldsymbol{Y}; \gamma_w^{-1})$.

Using the message derivation rules of loopy BP, the four messages defined



$\mu_{(i,j)\to i,t}$	message from node f_{ij} to node $\mathbf{u}_i$
$\nu_{(i,j)\to j,t}$	message from node f_{ij} to node $\mathbf{v}_j$
$\mu_{i\to(i,j),t}$	message from node $\mathbf{u}_i$ to node f_{ij}
$\nu_{j\to(i,j),t}$	message from node $\mathbf{v}_j$ to node f_{ij}

Figure 4.1: Factor graph associated to (4.3). The circles represent variable nodes and the squares represent factor nodes.

in Fig. 4.1, , are expressed as follows (where t stands for the iteration index):

$$\mu_{(i,j)\to i,t}\left(\boldsymbol{u}_i\right) \propto \int f(\boldsymbol{u}_i,\boldsymbol{v}_j)\ \nu_{j\to(i,j),t-1}\left(\boldsymbol{v}_j\right)\mathrm{d}\boldsymbol{v}_j, \tag{4.5a}$$

$$\mu_{i\to(i,j),t+1}\left(\boldsymbol{u}_i\right) \propto p_{\mathbf{u}}\left(\boldsymbol{u}_i\right)^\beta \prod_{l\neq j}\mu_{(i,l)\to i,t}\left(\boldsymbol{u}_i\right), \tag{4.5b}$$

$$\nu_{(i,j)\to j,t}\left(\boldsymbol{v}_j\right) \propto \int f(\boldsymbol{u}_i,\boldsymbol{v}_j)\ \mu_{i\to(i,j),t-1}\left(\boldsymbol{u}_i\right)\mathrm{d}\boldsymbol{u}_i, \tag{4.5c}$$

$$\nu_{j\to(i,j),t+1}\left(\boldsymbol{v}_j\right) \propto p_{\mathbf{v}}\left(\boldsymbol{v}_j\right)^\beta \prod_{k\neq i}\nu_{(k,j)\to j,t}\left(\boldsymbol{v}_j\right). \tag{4.5d}$$

Assuming $\mathbf{v}_l \sim \nu_{l\to(i,l),t}\left(\boldsymbol{v}_l\right)$ with mean $\widehat{\boldsymbol{v}}_{l\to(i,l),t}$ and covariance matrix $\beta^{-1}\boldsymbol{R}_{\mathbf{v},l\to(i,l),t}$, it was shown in [36] by virtue of the CLT that the product of incoming messages, $\prod_{l\neq j}\mu_{(i,l)\to i,t}\left(\boldsymbol{u}_i\right)$, to any variable node $\mathbf{u}_i$ from all factor nodes $\{f_{il}\}_{l\neq j}$ can be approximated by a Gaussian density with mean $\boldsymbol{b}_{\mathbf{u},i\to(i,j),t}$ and precision $\beta\boldsymbol{\Lambda}_{\mathbf{u},i\to(i,j),t}$:

$$\boldsymbol{b}_{\mathbf{u},i\to(i,j),t} \;=\; \gamma_w \sum_{l\neq j} y_{i,l}\, \widehat{\boldsymbol{v}}_{l\to(i,l),t}, \tag{4.6a}$$

$$\boldsymbol{\Lambda}_{\mathbf{u},i\to(i,j),t} \;=\; \gamma_w \sum_{l\neq j} \Big(\widehat{\boldsymbol{v}}_{l\to(i,l),t}\, \widehat{\boldsymbol{v}}^{\top}_{l\to(i,l),t} \;+\; \beta^{-1} \boldsymbol{R}_{\mathbf{v},l\to(i,l),t} - \gamma_w \; y^2_{i,l}\, \boldsymbol{R}_{\mathbf{v},l\to(i,l),t} \Big). \tag{4.6b}$$

In other words, by dropping the normalization factor that does not depend on $\boldsymbol{u}_i$, we have:

$$\prod_{l\neq j} \mu_{(i,l)\to i,t}\left(\boldsymbol{u}_i\right) \propto \exp\left(-\frac{\beta}{2} \boldsymbol{u}_i^{\top} \boldsymbol{\Lambda}_{\mathbf{u},i\to(i,j),t} \boldsymbol{u}_i + \beta\, \boldsymbol{u}_i^{\top} \boldsymbol{b}_{\mathbf{u},i\to(i,j),t} \right). \tag{4.7}$$

The inherent symmetry among the variable nodes $\mathbf{u}_i$ and $\mathbf{v}_j$ yields an equivalent Gaussian approximation for $\prod_{k\neq i} \nu_{(k,j)\to j,t}\left(\boldsymbol{v}_j\right)$ under the density of $\mathbf{u}_k \sim \mu_{k\to(k,j),t}\left(\boldsymbol{u}_k\right)$ with mean $\widehat{\boldsymbol{u}}_{k\to(k,j),t}$ and covariance matrix $\beta^{-1}\boldsymbol{R}_{\mathbf{u},k\to(k,j),t}$.

$$\prod_{k\neq i} \nu_{(k,j)\to j,t}\left(\boldsymbol{v}_j\right) \propto \exp\left(-\frac{\beta}{2} \boldsymbol{v}_j^{\top} \boldsymbol{\Lambda}_{\mathbf{v},j\to(i,j),t} \boldsymbol{v}_j + \beta\, \boldsymbol{v}_j^{\top} \boldsymbol{b}_{\mathbf{v},j\to(i,j),t} \right), \tag{4.8}$$

with

$$\boldsymbol{b}_{\mathbf{v},j\to(i,j),t} \;=\; \gamma_w \sum_{k\neq i} y_{k,j}\, \widehat{\boldsymbol{u}}_{k\to(k,j),t}, \tag{4.9a}$$

$$\boldsymbol{\Lambda}_{\mathbf{v},j\to(i,j),t} \;=\; \gamma_w \sum_{k\neq i} \Big(\widehat{\boldsymbol{u}}_{k\to(k,j),t}\, \widehat{\boldsymbol{u}}^{\top}_{k\to(k,j),t} + \beta^{-1} \boldsymbol{R}_{\mathbf{u},k\to(k,j),t} - \gamma_w\, y^2_{k,j}\, \boldsymbol{R}_{\mathbf{u},k\to(k,j),t} \Big). \tag{4.9b}$$

Pictorially, the messages given in (4.7) and (4.8) are shown in Fig. 4.2 as messages ① and ②. Note here that ① is the aggregate message that is collectively sent by all factor nodes $\{f_{ij'}\}_{j'=1,j'\neq j}^{M}$ to $\boldsymbol{u}_i$. Similarly ② is the aggregate message that is collectively sent by all factor nodes $\{f_{i'j}\}_{i'=1,i'\neq i}^{N}$ to $\boldsymbol{v}_j$. Moreover, the mean values, $\widehat{\boldsymbol{u}}_{i\to(i,j),t+1}$ and $\widehat{\boldsymbol{v}}_{j\to(i,j),t+1}$, as well as the covariance matrices, $\beta^{-1}\boldsymbol{R}_{\mathbf{u},i\to(i,j),t+1}$ and $\beta^{-1}\boldsymbol{R}_{\mathbf{v},j\to(i,j),t+1}$, of messages ③ and ④, respectively, are given in (4.16a)–(4.16d). To reduce the complexity in the number of computed messages, the $(i,j)-$dependent quantities in (4.6) and (4.9), are replaced by the following $(i,j)-$independent (i.e.,

broadcast) ones:

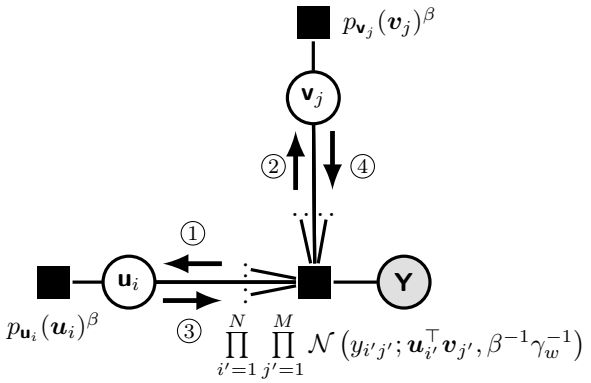
$$\boldsymbol{b}_{\mathbf{u},i,t} \;=\; \gamma_w \sum_l y_{i,l}\, \widehat{\boldsymbol{v}}_{l\to(i,l),t}, \tag{4.10a}$$

$$\boldsymbol{\Lambda}_{\mathbf{u},i,t} \;=\; \gamma_w \sum_l \Big(\widehat{\boldsymbol{v}}_{l\to(i,l),t}\, \widehat{\boldsymbol{v}}^{\top}_{l\to(i,l),t} \;+\; \big(\beta^{-1} - \gamma_w\, y^2_{i,l}\big)\, \boldsymbol{R}_{\mathbf{v}_l,t} \Big), \tag{4.10b}$$

$$\boldsymbol{b}_{\mathbf{v},j,t} \;=\; \gamma_w \sum_k y_{j,k}\, \widehat{\boldsymbol{u}}_{k\to(k,j),t}, \tag{4.11a}$$

$$\boldsymbol{\Lambda}_{\mathbf{v},j,t} \;=\; \gamma_w \sum_k \Big(\widehat{\boldsymbol{u}}_{k\to(k,j),t}\, \widehat{\boldsymbol{u}}^{\top}_{k\to(k,j),t} + \big(\beta^{-1} - \gamma_w y^2_{k,j}\big)\, \boldsymbol{R}_{\mathbf{u}_k,t} \Big). \tag{4.11b}$$

This follows from the approximation of the covariance matrices, $\beta^{-1}\boldsymbol{R}_{\mathbf{u},i\to(i,j),t}$



①	$\mathcal{N}(\boldsymbol{u}_i; \boldsymbol{\Lambda}^{-1}_{\mathbf{u},i\to(i,j),t}\boldsymbol{b}_{\mathbf{u},i\to(i,j),t}, \beta^{-1}\boldsymbol{\Lambda}^{-1}_{\mathbf{u},i\to(i,j),t})$
②	$\mathcal{N}(\boldsymbol{v}_j; \boldsymbol{\Lambda}^{-1}_{\mathbf{v},j\to(i,j),t}\boldsymbol{b}_{\mathbf{v},j\to(i,j),t}, \beta^{-1}\boldsymbol{\Lambda}^{-1}_{\mathbf{v},j\to(i,j),t})$

Figure 4.2: Explicit messages resulting from the CLT approximations shown here for a single cell of the entire factor graph depicted in Fig. 4.1.

and $\beta^{-1}\boldsymbol{R}_{\mathbf{v},j\to(i,j),t}$, involved in (4.6) and (4.9) by broadcast covariances, $\beta^{-1}\boldsymbol{R}_{\mathbf{u}_i,t}$ and $\beta^{-1}\boldsymbol{R}_{\mathbf{v}_j,t}$, respectively, with a vanishing error order $O(M^{-1})$ [36]. By recalling (4.16a)-(4.16d), the underlying broadcast means and covariances are given by:

$$\widehat{\boldsymbol{u}}_{i,t+1} \;=\; \mathbf{f}_{\mathsf{u}}(\boldsymbol{b}_{\mathbf{u},i,t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{u},i,t}), \tag{4.12}$$

$$\boldsymbol{R}_{\mathbf{u}_i,t+1} \;=\; \nabla_{\boldsymbol{b}_{\mathbf{u},i,t}} \mathbf{f}_{\mathsf{u}}(\boldsymbol{b}_{\mathbf{u},i,t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{u},i,t})^{\mathsf{T}}, \tag{4.13}$$

$$\widehat{\boldsymbol{v}}_{j,t+1} \;=\; \mathbf{f}_{\mathsf{v}}(\boldsymbol{b}_{\mathbf{v},j,t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{v},j,t}), \tag{4.14}$$

$$\boldsymbol{R}_{\mathbf{v}_j,t+1} \;=\; \nabla_{\boldsymbol{b}_{\mathbf{v},j,t}} \mathbf{f}_{\mathsf{v}}(\boldsymbol{b}_{\mathbf{v},j,t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{v},j,t})^{\mathsf{T}}. \tag{4.15}$$

The relationship between the (i,j) posterior means, $\widehat{\boldsymbol{u}}_{i\to(i,j),t}$ and $\widehat{\boldsymbol{v}}_{j\to(i,j),t}$, and

their broadcast versions, $\widehat{\boldsymbol{u}}_{i,t}$ and $\widehat{\boldsymbol{v}}_{j,t}$, through small Onsager correction terms of order $O(M^{-1/2})$ are given in (4.20) where the correction terms are taken into account during the calculation of $\boldsymbol{b}_{\mathbf{u},i,t}$ and $\boldsymbol{b}_{\mathbf{v},j,t}$ only. For the computation of $\boldsymbol{\Lambda}_{\mathbf{u},i,t}$ and $\boldsymbol{\Lambda}_{\mathbf{v},j,t}$, however, one replaces $\widehat{\boldsymbol{u}}_{i\to(i,j),t}$ by $\widehat{\boldsymbol{u}}_{i,t}$ and $\widehat{\boldsymbol{v}}_{j\to(i,j),t}$ by $\widehat{\boldsymbol{v}}_{j,t}$ after ignoring terms of vanishing order as $M, N \to \infty$.

4.1.3 From low-rank matrix reconstruction to BiG-VAMP

In this section, we review the main results of the prior work on *low-rank* matrix reconstruction in [36] at the detail needed for a comprehensive exposition of BiG-VAMP. We emphasize, however, the fact that borrowing those results does not restrict the proposed BiG-VAMP algorithm to the bilinear *low-rank* matrix recovery as is the case in [36].

The name temperature parameter of the variable β in Eq (4.3), is used in pure analogy to the role β plays in statistical physics [46]. We assume the priors, $p_{\mathbf{U}}(\boldsymbol{U})$ and $p_{\mathbf{V}}(\boldsymbol{V})$, on the unknown matrices of $\mathbf{U}$ and $\mathbf{V}$ to be row-wise separable, i.e., $p_{\mathbf{U}}(\boldsymbol{U}) = \prod_{i=1}^{N} p_{\mathbf{u}}(\boldsymbol{u}_i)$ and $p_{\mathbf{V}}(\boldsymbol{V}) = \prod_{j=1}^{M} p_{\mathbf{v}}(\boldsymbol{v}_j)$. The posterior mean, $\widehat{\boldsymbol{u}}_{i\to(i,j),t+1}$ [resp. $\widehat{\boldsymbol{v}}_{j\to(i,j),t+1}$], and covariance matrix, ${}^{\beta^{-1}}\boldsymbol{R}_{\mathbf{u},i\to(i,j),t+1}$ [resp. ${}^{\beta^{-1}}\boldsymbol{R}_{\mathbf{v},j\to(i,j),t+1}$] from the variable node $\boldsymbol{u}_i$ [resp. $\boldsymbol{v}_j$] to the factor node $f_{ij}(\boldsymbol{u}_i, \boldsymbol{v}_j)$ are given by (see 4.1.2):

$$\widehat{\boldsymbol{u}}_{i\to(i,j),t+1} = \mathsf{f}_{\mathsf{u}}(\boldsymbol{b}_{\mathbf{u},i\to(i,j),t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{u},i\to(i,j),t}), \tag{4.16a}$$

$$\widehat{\boldsymbol{v}}_{j\to(i,j),t+1} = \mathsf{f}_{\mathsf{v}}(\boldsymbol{b}_{\mathbf{v},j\to(i,j),t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{v},j\to(i,j),t}), \tag{4.16b}$$

$$\boldsymbol{R}_{\mathbf{u},i\to(i,j),t+1} = \nabla_{\boldsymbol{b}_{\mathbf{u},i\to(i,j),t}} \mathsf{f}_{\mathsf{u}}(\boldsymbol{b}_{\mathbf{u},i\to(i,j),t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{u},i\to(i,j),t})^{\mathsf{T}}, \tag{4.16c}$$

$$\boldsymbol{R}_{\mathbf{v},j\to(i,j),t+1} = \nabla_{\boldsymbol{b}_{\mathbf{v},j\to(i,j),t}} \mathsf{f}_{\mathsf{v}}(\boldsymbol{b}_{\mathbf{v},j\to(i,j),t}, \boldsymbol{\Lambda}^{-1}_{\mathbf{v},j\to(i,j),t})^{\mathsf{T}}, \tag{4.16d}$$

Here, the quantities $\boldsymbol{b}_{\mathbf{u},i\to(i,j),t}$, $\boldsymbol{\Lambda}_{\mathbf{u},i\to(i,j),t}$, $\boldsymbol{b}_{\mathbf{v},i\to(i,j),t}$, and $\boldsymbol{\Lambda}_{\mathbf{v},j\to(i,j),t}$ are defined in (4.10a),(4.10b),(4.11a), and (4.11b), respectively. Moreover, the denoising func-

tions $\mathbf{f}_{\mathsf{u}}(.,.)$ and $\mathbf{f}_{\mathsf{v}}(.,.)$ are given by:

$$\mathbf{f}_{\mathsf{u}}(\boldsymbol{b},\boldsymbol{\Lambda}^{-1}) \;=\; \frac{\displaystyle\int \boldsymbol{u}\, p_{\mathbf{u}}\left(\boldsymbol{u}\right)^{\beta} \mathcal{N}\big(\boldsymbol{u};\boldsymbol{\Lambda}^{-1}\boldsymbol{b},\beta^{-1}\boldsymbol{\Lambda}^{-1}\big)\mathrm{d}\boldsymbol{u}}{\displaystyle\int p_{\mathbf{u}}\left(\boldsymbol{u}\right)^{\beta} \mathcal{N}\big(\boldsymbol{u};\boldsymbol{\Lambda}^{-1}\boldsymbol{b},\beta^{-1}\boldsymbol{\Lambda}^{-1}\big)\mathrm{d}\boldsymbol{u}}, \tag{4.17}$$

$$\mathbf{f}_{\mathsf{v}}(\boldsymbol{b},\boldsymbol{\Lambda}^{-1}) \;=\; \frac{\displaystyle\int \boldsymbol{v}\, p_{\mathbf{v}}\left(\boldsymbol{v}\right)^{\beta} \mathcal{N}\big(\boldsymbol{v};\boldsymbol{\Lambda}^{-1}\boldsymbol{b},\beta^{-1}\boldsymbol{\Lambda}^{-1}\big)\mathrm{d}\boldsymbol{v}}{\displaystyle\int p_{\mathbf{v}}\left(\boldsymbol{v}\right)^{\beta} \mathcal{N}\big(\boldsymbol{v};\boldsymbol{\Lambda}^{-1}\boldsymbol{b},\beta^{-1}\boldsymbol{\Lambda}^{-1}\big)\mathrm{d}\boldsymbol{v}}, \tag{4.18}$$

in which the nabla operator, $\nabla_{\boldsymbol{x}}$, with respect to any $n-$dimensional vector, $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^{\mathsf{T}}$, is given by:

$$\nabla_{\boldsymbol{x}} \;=\; \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \cdots, \frac{\partial}{\partial x_n}\right]^{\mathsf{T}}. \tag{4.19}$$

To reduce the complexity in the number of computed messages, the $(i,j)-$dependent quantities involved in (4.16a)–(4.16d) are replaced by $(i,j)-$independent (i.e., broadcast) ones, namely $\boldsymbol{b}_{\mathbf{u},i,t}$, $\boldsymbol{\Lambda}_{\mathbf{u},i,t}$, $\boldsymbol{b}_{\mathbf{v},j,t}$, and $\boldsymbol{\Lambda}_{\mathbf{v},j,t}$ given in (4.10) and (4.11). The (i,j) posterior means, $\widehat{\boldsymbol{u}}_{i\to(i,j),t}$ and $\widehat{\boldsymbol{v}}_{j\to(i,j),t}$, are related to the their broadcast versions, $\widehat{\boldsymbol{u}}_{i,t}$ and $\widehat{\boldsymbol{v}}_{j,t}$, through small Onsager correction terms as follows:

$$\widehat{\boldsymbol{u}}_{i\to(i,j),t} \;\approx\; \widehat{\boldsymbol{u}}_{i,t} \;-\; \underbrace{\gamma_w\, y_{ij}\, \boldsymbol{R}_{\boldsymbol{u}_i,t}\, \widehat{\boldsymbol{v}}_{j,t-1}}_{\text{Onsager correction term on } \boldsymbol{u}_i}, \tag{4.20a}$$

$$\widehat{\boldsymbol{v}}_{j\to(i,j),t} \;\approx\; \widehat{\boldsymbol{v}}_{j,t} \;-\; \underbrace{\gamma_w\, y_{ij}\, \boldsymbol{R}_{\boldsymbol{v}_j,t}\, \widehat{\boldsymbol{u}}_{i,t-1}}_{\text{Onsager correction term on } \boldsymbol{v}_j}. \tag{4.20b}$$

Moreover, the algorithm in [36] also relies on the following low-SNR approximation:

$$y_{k,j}^2 \;\approx\; \mathbb{E}[y_{k,j}^2] \;\approx\; \gamma_w^{-1}, \tag{4.21}$$

which is used to compute both $\boldsymbol{b}_{\mathbf{u},i,t}$ in (4.10a) and $\boldsymbol{b}_{\mathbf{v},j,t}$ in (4.11a). We emphasize, however, the fact that the low-SNR regime is mainly convenient in the presence of very-low-rank structures with a fully observed matrix $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{V}^{\top} + \boldsymbol{W}$.

The limitation of this method, however, lies in the intractability of the multi-dimensional integrals in (4.17) and (4.18). In fact, they can be computed only for some specific priors, $p_{\mathbf{u}}(\cdot)$ and $p_{\mathbf{v}}(\cdot)$, such as Gaussian and/or assignment priors. It is impossible, for instance, to accommodate a binary prior on $\mathbf{u}_i$ and/or $\mathbf{v}_j$ since

the underlying integrals become combinatorial sums over 2^r terms. In this context, the fundamental novelties brought by the proposed BiG-VAMP algorithm consist of its combined abilities to handle:

- A broader class of practical applications which involve *high-rank* decompositions of structured matrices in addition to *low-rank* decompositions of (possibly) unstructured matrices w.r.t. the prior information on $\boldsymbol{U}$ and/or $\boldsymbol{V}$.
- General priors on both $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices owing to appropriate Gaussian approximation of the extrinsic information exchanged between the algorithm's constituent blocks.
- General separable output distributions, $p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y}|\boldsymbol{Z})$, in (4.1).

4.2 Quantized BiG-VAMP for discrete MC

Before delving into the derivation details, it is important to remember that the discrete MF/MC problem we are aiming to solve, which is given in the observation model (4), is a noiseless version of the generalized bilinear model formulated in (4.2), wherein, the generalized function $\phi(\cdot)$, is a cascade of the selection and quantization functions, $\phi(\cdot) = \phi_s \circ \phi_q(\cdot)$.

We emphasize, however, the fact that a dedicated customization of BiG-VAMP coined BiG-VAMP-MC, which runs iteratively according to the algorithmic steps of Algorithm 3. For better illustration, the block diagram of BiG-VAMP-MC is depicted in Fig. 4.3 whereby we show its different constituent blocks as they interact with the so-called Bi-LMMSE module through the extrinsic information. These modules are:

1. the two i.i.d. denoisers of $\boldsymbol{U}$ and $\boldsymbol{V}$ in red;
2. the approximate Bi-LMMSE module in purple, which relies on the Gaussian message passing;
3. the output denoiser module in green, which accounts for both the selection transformation $\phi_s(\cdot)$ and the quantization transformation $\phi_q(\cdot)$;
4. The EM learning module of the quantization thresholds is in yellow.

BiG-VAMP for MF with partially observed data runs iteratively according to the four algorithmic steps described in Algorithm 3:

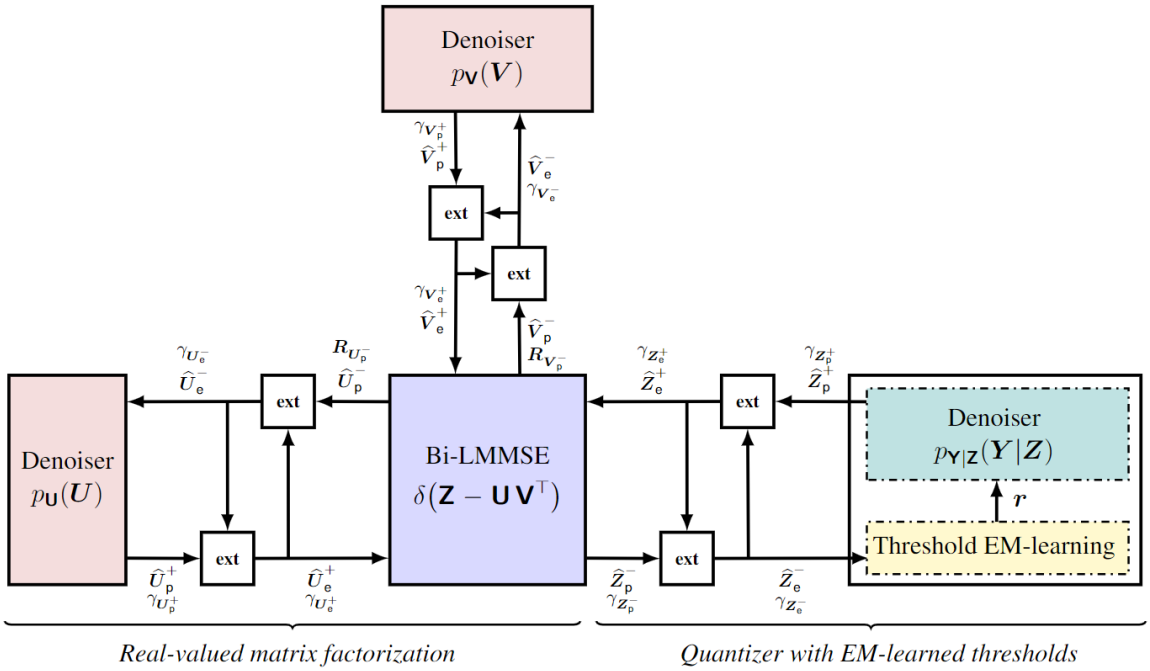


Figure 4.3: Block diagram of BiG-VAMP-MC with its five main modules: two denoising modules (MMSE or MAP) incorporating the prior information, $p_{\mathbf{U}}(\cdot)$ and $p_{\mathbf{V}}(\cdot)$, the bi-LMMSE module, the EM threshold learning module, and one output denoising module (MMSE or MAP) handling the observation model $p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y}|\boldsymbol{Z})$. The five modules exchange extrinsic information/messages calculated through the ext blocks, and $\delta(\cdot)$ denotes the Dirac delta distribution.

4.2.1 Denoising step

Inspired by the variable split technique used in VAMP construction seen in (3.3), and to remedy the intractable integrals during the evaluation of the densities ③ and ④ in Fig. 4.2, we proceed as follows. We rewrite the original posterior factorization in (4.3) by splitting $\mathbf{u}_i$ (resp. $\mathbf{v}_j$) into two identical variables with equality constraints in between, i.e., $\mathbf{u}_i^+ = \mathbf{u}_i^-$ (resp. $\mathbf{v}_j^+ = \mathbf{v}_j^-$) as seen in the factor graph depicted in Fig. 4.4. To handle the newly introduced equality constraints, the new variables $\mathbf{u}_i^+$, $\mathbf{u}_i^-$, $\mathbf{v}_j^+$, and $\mathbf{v}_j^-$ are rather regarded as processing nodes which exchange scalar messages/beliefs in the form of component-wise (i.e., decoupled) Gaussian densities. The beliefs provided by $\mathbf{u}_i^-$ and $\mathbf{v}_i^-$ on $\mathbf{u}_i^+$ and $\mathbf{v}_i^+$ (and vice versa) are known as the *extrinsic information* and are modelled by the Gaussian messages ①', ②', ③' and ④' in Fig. 4.4.

The decoupling in the messages on each side of the equality nodes results in two simple types of denoising functions, namely $\mathbf{f}_{\mathbf{u}}(\cdot,\cdot)$ [resp. $\mathbf{f}_{\mathbf{v}}(\cdot,\cdot)$] and $\mathbf{g}_{\mathbf{u}}(\cdot,\cdot)$ [resp. $\mathbf{g}_{\mathbf{v}}(\cdot,\cdot)$] to recover $\boldsymbol{u}_i^-$ (resp. $\boldsymbol{v}_j^-$) and $\boldsymbol{u}_i^+$ (resp. $\boldsymbol{v}_j^+$). Such decoupled message

Algorithm 3 BiG-VAMP-MC

Require : Matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times M}$; EM stoping precision ϵ_{EM} or number of EM iterations k_{EM}; parameter precision tolerance ($\xi = 10^{-6}$); maximum number of iterations ($T_{\max}$); two denoisers $\mathbf{g}_{\mathsf{u}}(\cdot)$ and $\mathbf{g}_{\mathsf{v}}(\cdot)$; initial quantization thresholds $\widehat{\boldsymbol{r}}_0^1 = [r_1^1, \ldots, r_{l-1}^1]$

Initialize

$t \leftarrow 1$

▷ posterior means, covariances and precisions

$\widehat{\boldsymbol{U}}_{\mathsf{p},1}^-$, $\widehat{\boldsymbol{V}}_{\mathsf{p},1}^-$, $\widehat{\boldsymbol{Z}}_{\mathsf{p},1}^-$, $\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,1}$, $\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,1}$, $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^-,1}$

$\widehat{\boldsymbol{U}}_{\mathsf{p},1}^+$, $\widehat{\boldsymbol{V}}_{\mathsf{p},1}^+$, $\widehat{\boldsymbol{Z}}_{\mathsf{p},1}^+$, $\gamma_{\boldsymbol{U}_{\mathsf{p}}^-,1}$, $\gamma_{\boldsymbol{V}_{\mathsf{p}}^-,1}$, $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^+,1}$

▷ extrinsic means and precisions

$\widehat{\boldsymbol{U}}_{\mathsf{e},1}^-$, $\widehat{\boldsymbol{V}}_{\mathsf{e},1}^-$, $\widehat{\boldsymbol{Z}}_{\mathsf{e},1}^-$, $\gamma_{\boldsymbol{U}_{\mathsf{e}}^-,1}$, $\gamma_{\boldsymbol{V}_{\mathsf{e}}^-,1}$, $\gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,1}$

$\widehat{\boldsymbol{U}}_{\mathsf{e},0}^+$, $\widehat{\boldsymbol{V}}_{\mathsf{e},0}^+$, $\widehat{\boldsymbol{U}}_{\mathsf{e},1}^+$, $\widehat{\boldsymbol{V}}_{\mathsf{e},1}^+$, $\widehat{\boldsymbol{Z}}_{\mathsf{e},1}^+$, $\gamma_{\boldsymbol{U}_{\mathsf{e}}^+,1}$, $\gamma_{\boldsymbol{V}_{\mathsf{e}}^+,1}$, $\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,1}$

▷ means and precisions of the Bi-LMMSE step

$\boldsymbol{B}_{\boldsymbol{U},1}$, $\boldsymbol{\Lambda}_{\boldsymbol{U},1}$, $\boldsymbol{B}_{\boldsymbol{V},1}$, $\boldsymbol{\Lambda}_{\boldsymbol{V},1}$

repeat

I. Approximate Bi-LMMSE step

▷ Compute the approximated quantities related to the Bi-LMMSE mean and covariances of $\boldsymbol{U}$

$\boldsymbol{B}_{\boldsymbol{U},t} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\big(\widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+ \widehat{\boldsymbol{V}}_{\mathsf{p},t}^- - M\,\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\,\widehat{\boldsymbol{U}}_{\mathsf{p},t-1}^-\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t}\,\langle \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+ \odot \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+\rangle\big)$

$\boldsymbol{\Lambda}_{\boldsymbol{U},t} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\big(\widehat{\boldsymbol{V}}_{\mathsf{p},t}^{-\top}\widehat{\boldsymbol{V}}_{\mathsf{p},t}^- + \frac{M}{\beta}\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t} - M\,\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t}\,\langle \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+ \odot \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+\rangle\big)$

▷ Compute the approximated quantities related to the Bi-LMMSE mean and covariances of $\boldsymbol{V}$

$\boldsymbol{B}_{\boldsymbol{V},t} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\big(\widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+ \widehat{\boldsymbol{U}}_{\mathsf{p},t}^- - N\,\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\,\widehat{\boldsymbol{V}}_{\mathsf{p},t-1}^-\,\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t}\,\langle \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+ \odot \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+\rangle\big)$

$\boldsymbol{\Lambda}_{\boldsymbol{V},t} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\big(\widehat{\boldsymbol{U}}_{\mathsf{p},t}^{-\top}\widehat{\boldsymbol{U}}_{\mathsf{p},t}^- + \frac{N}{\beta}\,\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t} - N\,\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\,\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t}\,\langle \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+ \odot \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+\rangle\big)$

▷ Update the posterior statistics $\widehat{\boldsymbol{U}}_{\mathsf{p},t}^-$, $\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t}$, $\widehat{\boldsymbol{U}}_{\mathsf{p},t}^-$, $\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t}$

$\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1} = (\gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t}\,\boldsymbol{I} + \boldsymbol{\Lambda}_{\boldsymbol{U},t})^{-1}$

$\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^- = (\boldsymbol{B}_{\boldsymbol{U},t} + \gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t}\widehat{\boldsymbol{U}}_{\mathsf{e},t}^+)\,\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1}$

$\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1} = (\gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t}\,\boldsymbol{I} + \boldsymbol{\Lambda}_{\boldsymbol{V},t})^{-1}$

$\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^- = (\boldsymbol{B}_{\boldsymbol{V},t} + \gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t}\widehat{\boldsymbol{V}}_{\mathsf{e},t}^+)\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1}$

II. Denoising step

▷ Update the extrinsic statistics $\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^-$, $\gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}$, $\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^-$, $\gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1}$

$\gamma_{\boldsymbol{U}_{\mathsf{p}}^-,t+1} = (\frac{1}{r}\mathrm{Tr}(\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1}))^{-1}$

$\gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1} = \gamma_{\boldsymbol{U}_{\mathsf{p}}^-,t+1} - \gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t}$

$\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^- = \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}^{-1}(\gamma_{\boldsymbol{U}_{\mathsf{p}}^-,t+1}\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^- - \gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t}\widehat{\boldsymbol{U}}_{\mathsf{e},t}^+)$

$\gamma_{\boldsymbol{V}_{\mathsf{p}}^-,t+1} = (\frac{1}{r}\mathrm{Tr}(\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1}))^{-1}$

$\gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1} = \gamma_{\boldsymbol{V}_{\mathsf{p}}^-,t+1} - \gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t}$

$\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^- = \gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1}^{-1}(\gamma_{\boldsymbol{V}_{\mathsf{p}}^-,t+1}\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^- - \gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t}\widehat{\boldsymbol{V}}_{\mathsf{e},t}^+)$

▷ Denoising the rows $\boldsymbol{u}_{i,t+1}^-$ of $\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}$

compute $\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^+$ such that every row $\boldsymbol{u}_{i,t+1}^+ = \mathbf{g}_{\mathsf{u}}(\boldsymbol{u}_{i,t+1}^-, \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}^{-1}; \beta)$, $\forall i$

$\gamma_{\boldsymbol{U}_{\mathsf{p}}^+,t+1} = \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}\left(\frac{1}{N}\sum_{i=1}^N \langle \mathbf{g}'_{\mathsf{u}}(\boldsymbol{u}_{i,t+1}^-, \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}^{-1}; \beta)\rangle\right)^{-1}$

▷ Denoising the columns $\boldsymbol{v}_{j,t+1}^-$ of $\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^-$

compute $\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^+$ such that every column $\boldsymbol{v}_{j,t+1}^+ = \mathbf{g}_{\mathsf{v}}(\boldsymbol{v}_{j,t+1}^-, \gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1}^{-1}; \beta)$, $\forall j$

$\gamma_{\boldsymbol{V}_{\mathsf{p}}^+,t+1} = \gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1}\left(\frac{1}{M}\sum_{j=1}^M \langle \mathbf{g}'_{\mathsf{v}}(\boldsymbol{v}_{j,t+1}^-, \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}^{-1}; \beta)\rangle\right)^{-1}$

▷ update the extrinsic statistics $\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^+$, $\gamma_{\boldsymbol{U}_{\mathsf{e},t+1}^+}$, $\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^+$, $\gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t+1}$

$\gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t+1} = \gamma_{\boldsymbol{U}_{\mathsf{p}}^+,t+1} - \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1}$

$\gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t+1} = \gamma_{\boldsymbol{V}_{\mathsf{p}}^+,t+1} - \gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1}$

$\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^+ = \gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t+1}^{-1}(\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^+\gamma_{\boldsymbol{U}_{\mathsf{p}}^+,t+1} - \widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^-\gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t+1})$

$\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^+ = \gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t+1}^{-1}(\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^+\gamma_{\boldsymbol{V}_{\mathsf{p}}^+,t+1} - \widehat{\boldsymbol{V}}_{\mathsf{e}}^-\gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t+1})$

III. Thresholds learning using EM algorithm:

repeat

The E-step: compute the expectation of (4.42) over the posterior (4.52) using $\widehat{\boldsymbol{r}}_0^t = [r_1^t, \ldots, r_{l-1}^t]^\top$

The M-step solve the maximization problem seen in (4.53) by using the gradient descent algorithm and update $\widehat{\boldsymbol{r}}_k^t$

until $\|\widehat{\boldsymbol{r}}_{k+1}^t - \widehat{\boldsymbol{r}}_k^t\|_{\mathrm{F}} \le \epsilon_{\text{EM}}$ or $(k > k_{\text{EM}})$

update $\widehat{\boldsymbol{r}}_0^{t+1} = \widehat{\boldsymbol{r}}_k^t$

IV. the quantization denoiser

▷ Compute the posterior statistics $\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^-$ and $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^-,t+1}$

$\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^- = \widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^-\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^{-\top} + \frac{\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}}{\beta}\widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+\,\mathrm{Tr}(\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1}\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1}^\top)$

$\gamma_{\boldsymbol{Z}_{\mathsf{p}}^-,t+1} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t} + MN\,\mathrm{Tr}(\frac{MN}{\beta}\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1}\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1}^\top + N\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1}\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^{-\top}\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^- + M\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1}\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^{-\top}\widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^-)^{-1}$

▷ Compute the extrinsic statistics $\widehat{\boldsymbol{Z}}_{\mathsf{e},t+1}^-$ and $\gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1}$

$\gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1} = \gamma_{\boldsymbol{Z}_{\mathsf{p}}^-,t+1} - \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}$

$\widehat{\boldsymbol{Z}}_{\mathsf{e},t+1}^- = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1}^{-1}(\widehat{\boldsymbol{Z}}_{\mathsf{p},t}^-\gamma_{\boldsymbol{Z}_{\mathsf{p}}^-,t+1} - \widehat{\boldsymbol{Z}}_{\mathsf{e},t}^+\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t})$

▷ Compute the posterior and extrinsic statistics $\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^+$, $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^+,t+1}$, $\widehat{\boldsymbol{Z}}_{\mathsf{e},t+1}^+$, $\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t+1}$

compute the discrete mean $\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^+$ and $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^+,t+1}$ element-wise using $\widehat{\boldsymbol{\theta}}_0^{t+1}$ like described in (4.36),(4.37),(4.38a) and (4.38b)

$\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t+1} = \gamma_{\boldsymbol{Z}_{\mathsf{p}}^+,t+1} - \gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1}$

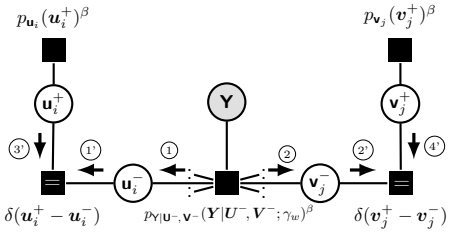
$\widehat{\boldsymbol{Z}}_{\mathsf{e},t+1}^+ = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t+1}(\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^+\gamma_{\boldsymbol{Z}_{\mathsf{p}}^+,t+1} - \widehat{\boldsymbol{Z}}_{\mathsf{e},t+1}^-\gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1})$

$t \leftarrow t+1$

until $\left(\|\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^+ - \widehat{\boldsymbol{U}}_{\mathsf{p},t}^+\|_{\mathrm{F}}^2 + \|\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^+ - \widehat{\boldsymbol{V}}_{\mathsf{p},t}^+\|_{\mathrm{F}}^2\right) \le \xi\left(\|\widehat{\boldsymbol{U}}_{\mathsf{p},t}^+\|_{\mathrm{F}}^2 + \|\widehat{\boldsymbol{V}}_{\mathsf{p},t}^+\|_{\mathrm{F}}^2\right)$ or $\left(t > T_{\max}\right)$

return $\widehat{\boldsymbol{U}}_{\mathsf{p},T_{\max}+1}^+$, $\widehat{\boldsymbol{V}}_{\mathsf{p},T_{\max}+1}^+$

passing is made possible by ignoring the off-diagonal elements of the error covariance matrices calculated on the side of $\mathbf{u}_i^-$ and $\mathbf{v}_j^-$ nodes. Although the integrals of the denoising functions, $\mathbf{f}_{\mathsf{u}}(\cdot,\cdot)$ and $\mathbf{f}_{\mathsf{v}}(\cdot,\cdot)$, in (4.17) and (4.18) are still required



①	$\mathcal{N}(\boldsymbol{u}_i^-; \boldsymbol{\Lambda}^{-1}_{\mathbf{u}_i^-,t}\boldsymbol{b}_{\mathbf{u}_i^-,t}, \beta^{-1}\boldsymbol{\Lambda}^{-1}_{\mathbf{u}_i^-,t})$
①'	$\mathcal{N}(\boldsymbol{u}_i^+; \widehat{\boldsymbol{u}}^-_{i,\mathsf{e},t+1}, \beta^{-1}\gamma^{-1}_{\mathbf{u}^-_{i,\mathsf{e}},t+1}\boldsymbol{I})$
②	$\mathcal{N}(\boldsymbol{v}_j^-; \boldsymbol{\Lambda}^{-1}_{\mathbf{v}_i^-,t}\boldsymbol{b}_{\mathbf{v}_j^-,t}, \beta^{-1}\boldsymbol{\Lambda}^{-1}_{\mathbf{v}_j^-,t})$
②'	$\mathcal{N}(\boldsymbol{v}_j^+; \widehat{\boldsymbol{v}}^-_{j,\mathsf{e},t+1}, \beta^{-1}\gamma^{-1}_{\mathbf{v}^-_{j,\mathsf{e}},t+1}\boldsymbol{I})$
③'	$\mathcal{N}(\boldsymbol{u}_i^-; \widehat{\boldsymbol{u}}^+_{i,\mathsf{e},t}, \beta^{-1}\gamma^{-1}_{\mathbf{u}^+_{i,\mathsf{e}},t}\boldsymbol{I})$
④'	$\mathcal{N}(\boldsymbol{v}_j^-; \widehat{\boldsymbol{v}}^+_{j,\mathsf{e},t}, \beta^{-1}\gamma^{-1}_{\mathbf{v}^+_{j,\mathsf{e}},t}\boldsymbol{I})$

Figure 4.4: Factor graph and the associated messages under generalized priors on $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices along with the Gaussian approximations for the extrinsic information (as reflected by the index e) that handles the equality constraints.

for Bi-VAMP, they are now evaluated in closed form after replacing the actual priors, $p_{\mathbf{u}}(\cdot)$ and $p_{\mathbf{v}}(\cdot)$, with the extrinsic Gaussian messages ③' and ④', respectively. Not considering the off-diagonal elements of the covariance matrices is what we call *the Gaussian approximation of priors* inspired by the turbo code principle. After computing the posterior estimates and the associated common precision (i.e., $\gamma_{\boldsymbol{U}^-_{\mathsf{p}},t+1}$ and $\gamma_{\boldsymbol{V}^-_{\mathsf{p}},t+1}$), each processing node $\mathbf{u}_i^-$ (resp. $\mathbf{v}_j^-$) subtracts the contribution of its incoming extrinsic message ③' (resp. ④') before returning the extrinsic messages ①' (resp. ②') to the other side of the equality node. Formally speaking, this amounts to approximating the posterior messages by Gaussian distributions with the same means and variances (i.e., $\mathcal{N}(\boldsymbol{u}_i^+; \widehat{\boldsymbol{u}}^-_{i,\mathsf{p},t+1}, \beta^{-1}\gamma^{-1}_{\boldsymbol{U}^-_{\mathsf{p}},t+1}\boldsymbol{I})$ and $\mathcal{N}(\boldsymbol{v}_j^+; \widehat{\boldsymbol{v}}^-_{j,\mathsf{p},t+1}, \beta^{-1}\gamma^{-1}_{\boldsymbol{V}^-_{\mathsf{p}},t+1}\boldsymbol{I})$) and extracting the extrinsic messages as follows:

$$\mathcal{N}(\boldsymbol{u}_i^+; \widehat{\boldsymbol{u}}^-_{i,\mathsf{e},t+1}, \beta^{-1}\gamma^{-1}_{\boldsymbol{U}^-_{\mathsf{e}},t+1}\boldsymbol{I}) \propto \frac{\mathcal{N}(\boldsymbol{u}_i^+; \widehat{\boldsymbol{u}}^-_{i,\mathsf{p},t+1}, \frac{1}{\beta}\gamma^{-1}_{\boldsymbol{U}^-_{\mathsf{p}},t+1}\boldsymbol{I})}{\mathcal{N}(\boldsymbol{u}_i^+; \widehat{\boldsymbol{u}}^+_{i,\mathsf{e},t}, \frac{1}{\beta}\gamma^{-1}_{\boldsymbol{U}^+_{\mathsf{e}},t}\boldsymbol{I})},$$

$$\mathcal{N}(\boldsymbol{v}_j^+; \widehat{\boldsymbol{v}}^-_{j,\mathsf{e},t+1}, \beta^{-1}\gamma^{-1}_{\boldsymbol{V}^-_{\mathsf{e}},t+1}\boldsymbol{I}) \propto \frac{\mathcal{N}(\boldsymbol{v}_j^+; \widehat{\boldsymbol{v}}^-_{j,\mathsf{p},t+1}, \frac{1}{\beta}\gamma^{-1}_{\boldsymbol{V}^-_{\mathsf{p}},t+1}\boldsymbol{I})}{\mathcal{N}(\boldsymbol{v}_j^+; \widehat{\boldsymbol{v}}^+_{j,\mathsf{e},t}, \frac{1}{\beta}\gamma^{-1}_{\boldsymbol{V}^+_{\mathsf{e}},t}\boldsymbol{I})}.$$

Given this extrinsic information and under separable priors $p_{\mathbf{u}}(\cdot)$ and $p_{\mathbf{v}}(\cdot)$, the denoising functions, $\mathbf{g}_{\mathsf{u}}(\cdot,\cdot)$, and $\mathbf{g}_{\mathsf{v}}(\cdot,\cdot)$, used to estimate $\boldsymbol{u}_i^+$ and $\boldsymbol{v}_j^+$, along with

their respective divergences, $\mathbf{g}'_{\mathsf{u}}(\cdot,\cdot)$ and $\mathbf{g}'_{\mathsf{v}}(\cdot,\cdot)$ in lines 18–21 of Algorithm 3 are given by:

$$\mathbf{g}_{\mathsf{u}}(\widehat{\boldsymbol{u}},\gamma_{\boldsymbol{U}}^{-1}) = \frac{\int \boldsymbol{u}\, p_{\mathbf{u}}(\boldsymbol{u})^{\beta}\, \mathcal{N}(\boldsymbol{u};\widehat{\boldsymbol{u}},\beta^{-1}\gamma_{\boldsymbol{U}}^{-1}\boldsymbol{I})\, d\boldsymbol{u}}{\int p_{\mathbf{u}}(\boldsymbol{u})^{\beta}\, \mathcal{N}(\boldsymbol{u};\widehat{\boldsymbol{u}},\beta^{-1}\gamma_{\boldsymbol{U}}^{-1}\boldsymbol{I})\, d\boldsymbol{u}}, \tag{4.23}$$

$$\mathbf{g}_{\mathsf{v}}(\widehat{\boldsymbol{v}},\gamma_{\boldsymbol{V}}^{-1}) = \frac{\int \boldsymbol{v}\, p_{\mathbf{v}}(\boldsymbol{v})^{\beta}\, \mathcal{N}(\boldsymbol{v};\widehat{\boldsymbol{v}},\beta^{-1}\gamma_{\boldsymbol{V}}^{-1}\boldsymbol{I})\, d\boldsymbol{v}}{\int p_{\mathbf{v}}(\boldsymbol{v})^{\beta}\, \mathcal{N}(\boldsymbol{v};\widehat{\boldsymbol{v}},\beta^{-1}\gamma_{\boldsymbol{V}}^{-1}\boldsymbol{I})\, d\boldsymbol{v}}, \tag{4.24}$$

$$\big[\mathbf{g}'_{\mathsf{u}}(\widehat{\boldsymbol{u}},\gamma_{\boldsymbol{U}}^{-1})\big]_{\ell} = \frac{\partial\big[\mathbf{g}_{\mathsf{u}}(\widehat{\boldsymbol{u}},\gamma_{\boldsymbol{U}}^{-1})\big]_{\ell}}{\partial\big[\widehat{\boldsymbol{u}}\big]_{\ell}},\ \ell=1,\ldots,r, \tag{4.25}$$

$$\big[\mathbf{g}'_{\mathsf{v}}(\widehat{\boldsymbol{v}},\gamma_{\boldsymbol{V}}^{-1})\big]_{\ell} = \frac{\partial\big[\mathbf{g}_{\mathsf{v}}(\widehat{\boldsymbol{v}},\gamma_{\boldsymbol{V}}^{-1})\big]_{\ell}}{\partial\big[\widehat{\boldsymbol{v}}\big]_{\ell}},\quad \ell=1,\ldots,r. \tag{4.26}$$

In essence, $\widehat{\boldsymbol{u}}^{+}_{i,\mathsf{p},t+1} = \mathbf{g}_{\mathsf{u}}(\widehat{\boldsymbol{u}}^{-}_{i,\mathsf{e},t+1},\gamma^{-1}_{\boldsymbol{U}^{-}_{\mathsf{e}},t+1})$ and $\widehat{\boldsymbol{v}}^{+}_{j,\mathsf{p},t+1} = \mathbf{g}_{\mathsf{v}}(\widehat{\boldsymbol{v}}^{-}_{j,\mathsf{e},t+1},\gamma^{-1}_{\boldsymbol{V}^{-}_{\mathsf{e}},t+1})$ are the posterior means of $\mathbf{u}_i^{+}$ and $\mathbf{v}_j^{+}$, respectively. In addition, their common posterior precisions updated in lines 19 and 21 of Algorithm 3 are given by:

$$\gamma_{\boldsymbol{U}^{+}_{\mathsf{p}},t+1} = \gamma_{\boldsymbol{U}^{-}_{\mathsf{e}},t+1}\left(\frac{1}{N}\sum_{i=1}^{N}\big\langle \mathbf{g}'_{\mathsf{u}}(\widehat{\boldsymbol{u}}^{-}_{i,\mathsf{e},t+1},\gamma^{-1}_{\boldsymbol{U}^{-}_{\mathsf{e}},t+1})\big\rangle\right)^{-1}, \tag{4.27}$$

$$\gamma_{\boldsymbol{V}^{+}_{\mathsf{p}},t+1} = \gamma_{\boldsymbol{V}^{-}_{\mathsf{e}},t+1}\left(\frac{1}{M}\sum_{j=1}^{M}\big\langle \mathbf{g}'_{\mathsf{v}}(\widehat{\boldsymbol{v}}^{-}_{j,\mathsf{e},t+1},\gamma^{-1}_{\boldsymbol{U}^{-}_{\mathsf{e}},t+1})\big\rangle\right)^{-1}. \tag{4.28}$$

4.2.2 Approximate Bi-LMMSE step

The goal of the Bi-LMMSE module is to reconstruct two random matrices $\mathbf{U} = (\mathbf{u}_1,\ldots,\mathbf{u}_N)^{\top} \in \mathbb{R}^{N\times R}$ and $\mathbf{V} = (\mathbf{v}_1,\ldots,\mathbf{v}_M)^{\top} \in \mathbb{R}^{M\times R}$ from a noisy update of $\boldsymbol{Z}$, $\widehat{\boldsymbol{Z}}^{+}_{\mathsf{e}} = \boldsymbol{Z} + \mathcal{N}(\boldsymbol{Z}_e;\boldsymbol{0},\gamma^{-1}_{\boldsymbol{Z}^{+}_{\mathsf{e}}}\boldsymbol{I})$, which is provided by the discrete denoiser from the previous iteration. Given some common priors, $p_{\mathbf{u}}(\boldsymbol{u}_i)$ and $p_{\mathbf{v}}(\boldsymbol{v}_i)$, on the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$, respectively, the goal is find the following posterior distribution:

$$p_{\mathbf{u},\mathbf{v}|\widehat{\mathbf{Z}}^{+}_{\mathsf{e}}}(\boldsymbol{u}_i,\boldsymbol{v}_j|\widehat{\boldsymbol{Z}}^{+}_{\mathsf{e}};\gamma^{-1}_{\boldsymbol{Z}^{+}_{\mathsf{e}}},\beta) \propto p_{\widehat{\mathbf{z}}^{+}_{ij,\mathsf{e}}|\mathbf{u},\mathbf{v}}\left(\widehat{z}^{+}_{ij,\mathsf{e}}|\boldsymbol{u}_i,\boldsymbol{v}_j;\gamma^{-1}_{\boldsymbol{Z}^{+}_{\mathsf{e}}}\right)^{\beta} p_{\mathbf{u}}(\boldsymbol{u}_i)^{\beta}\, p_{\mathbf{v}}(\boldsymbol{v}_j)^{\beta}, \tag{4.29}$$

To approximate (4.29), the message derivation rules of BP, along with the CLT. Such approximation was introduced in [36] for specific priors, $p_{\mathbf{U}}(\cdot)$ and $p_{\mathbf{V}}(\cdot)$ such as the assignment prior for clustering, and was recently generalized in [1] to treat

arbitrary i.i.d. priors owing to the expectation propagation framework.

To that end, we split $\mathbf{u}_i$ (and $\mathbf{v}_j$ and z_{ij}) into two identical variables with equality constraints, i.e., $\mathbf{u}_i^+ = \mathbf{u}_i^-$ (and $\mathbf{v}_j^+ = \mathbf{v}_j^-$ and $\mathsf{z}_{ij}^+ = \mathsf{z}_{ij}^-$). This allows us to rewrite (4.3) and obtain the following equivalent factorization:

$$p_{\mathbf{u}_i^+,\mathbf{u}_i^-,\mathbf{v}_j^+,\mathbf{v}_j^-|\widehat{\mathbf{Z}}_{\mathsf{e}}^+}\left(\boldsymbol{u}_i^+,\boldsymbol{u}_i^-,\boldsymbol{v}_j^+,\boldsymbol{v}_j^-|\widehat{\boldsymbol{Z}}_{\mathsf{e}}^+;\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+}^{-1},\beta\right) \propto p_{\widehat{\mathsf{z}}_{ij,\mathsf{e}}^+|\mathsf{z}_{ij}^+}\left(\widehat{z}_{ij,\mathsf{e}}^+|z_{ij}^+\right)\,\delta(z_{ij}^+ - z_{ij}^-)$$
$$\times\; p_{\mathsf{z}_{ij}^-|\mathbf{u}_i^-,\mathbf{v}_j^-}\left(z_{ij}^-|\boldsymbol{u}_i^-,\boldsymbol{v}_j^-;\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+}^{-1}\right)^{\beta}\,\delta(\boldsymbol{u}_i^- - \boldsymbol{u}_i^+)\times p_{\mathbf{u}}(\boldsymbol{u}_i^+)^{\beta}\,\delta(\boldsymbol{v}_j^- - \boldsymbol{v}_j^+)\,p_{\mathbf{v}}(\boldsymbol{v}_j^+)^{\beta}\,, \tag{4.30}$$

The factor graph associated to (4.30) is depicted in Fig. 4.5.

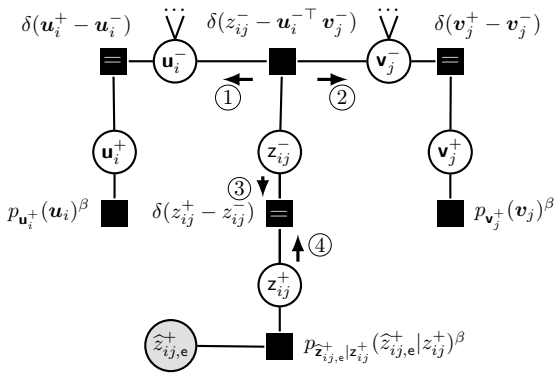


Figure 4.5: Factor graph for the generalized bilinear signal recovery problem with auxiliary variable nodes and equality constraints.

In contrast to BiG-AMP, which approximates *each* factor-to-variable scalar-valued node messages ① and ② separately, BiG-VAMP approximates in lines 4–7 *the product* of all incoming messages ① and ② to any vector-valued variable node, $\mathbf{u}_i^-$ and $\mathbf{v}_j^-$, with a Gaussian density $\mathcal{N}(\boldsymbol{u}_i^-;\boldsymbol{\Lambda}_{\mathbf{u}_i}^{-1}\boldsymbol{b}_{\mathbf{u}_i},\beta^{-1}\boldsymbol{\Lambda}_{\mathbf{u}_i}^{-1})$ and $\mathcal{N}(\boldsymbol{v}_j^-;\boldsymbol{\Lambda}_{\mathbf{v}_j}^{-1}\boldsymbol{b}_{\mathbf{v}_j},\beta^{-1}\boldsymbol{\Lambda}_{\mathbf{v}_j}^{-1})$, respectively. Using the latter densities, the posterior densities of the unknown matrices $\mathbf{U}$ and $\mathbf{V}$ with means $\widehat{\boldsymbol{U}}_{\mathsf{p}}^-$ and $\widehat{\boldsymbol{V}}_{\mathsf{p}}^-$, and posterior covariances $\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-}$ and $\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-}$ are computed in lines 8–11. Adopting the approximation of each message in BiG-AMP would reduce the necessity for tracking the covariance matrices of all incoming messages, but it leads to less accurate Bi-LMMSE estimation for discrete-valued priors, as shown in Section 5.3.

4.2.3 The quantization denoiser

The factor graph in this case as shown in Fig. (4.5), has a new variable node and a new factor node which corresponds to the new linear function extension. Once the quantization levels are learned and updated, We again resort to

the expectation propagation (aka turbo) principle to approximate the posterior messages in Fig. 4.5 by Gaussian distributions, $\mathcal{N}(z_{ij}^{-};\widehat{z}_{ij,\mathsf{e},t+1}^{-},\beta^{-1}\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{-},t+1}^{-1})$ and $\mathcal{N}(z_{ij}^{+};\widehat{z}_{ij,\mathsf{e},t}^{+},\beta^{-1}\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{+},t}^{-1})$, respectively, whose means and variances are calculated in the sequel. After that, this denoiser handles both element-wise output quantization and selection transformations, denoted respectively, $\phi_q(\cdot)$ and $\phi_s(\cdot)$. So, by defining $\mathsf{z}_{ij}^{-} \triangleq \mathbf{u}_i^{-\top}\mathbf{v}_j^{-}$, the quantized denoiser (the green module in Fig. 4.3) computes the posterior mean and variance, $\widehat{z}_{ij,\mathsf{p}}^{+}$ and $\gamma_{\mathsf{z}_{ij,\mathsf{p}}^{+}}^{-1}$, of $\mathsf{z}_{ij}^{+} \triangleq \mathbf{u}_i^{+\top}\mathbf{v}_j^{+}$ under the element-wise output likelihood channel scalar transformation, $p_{\mathsf{y}_{ij}|\mathsf{z}_{ij}^{+}}(y_{ij}|z_{ij}^{+})$ as follows:

$$\widehat{z}_{ij,\mathsf{p},t+1}^{+} = g_{\mathsf{z}}(y_{ij},\widehat{z}_{ij,\mathsf{e},t+1}^{-},\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{-},t+1}^{-1}), \tag{4.31}$$

$$\gamma_{\mathsf{z}_{ij,\mathsf{p}}^{+},t+1}^{-1} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^{-},t+1}^{-1}\frac{\partial g_{\mathsf{z}}\big(y_{ij},\widehat{z}_{ij,\mathsf{e},t+1}^{-},\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{-},t+1}^{-1}\big)}{\partial\widehat{z}_{ij,\mathsf{e},t+1}^{-}}, \tag{4.32}$$

with $g_{\mathsf{z}}(y,\widehat{z},\gamma_z^{-1})$ being the following scalar denoising function:

$$g_{\mathsf{z}}(y,\widehat{z},\gamma_z^{-1}) = \frac{\int_{-\infty}^{+\infty} z\,\mathcal{N}(z;\widehat{z},\beta^{-1}\gamma_z^{-1})\,p_{\mathsf{y}|\mathsf{z}}(y|z)^{\beta}\,dz}{\int_{-\infty}^{+\infty}\mathcal{N}(z;\widehat{z},\beta^{-1}\gamma_z^{-1})\,p_{\mathsf{y}|\mathsf{z}}(y|z)^{\beta}\,dz}. \tag{4.33}$$

Given the density of message (7) and under the linear constraint $\boldsymbol{Z}=\boldsymbol{U}\boldsymbol{V}^{\top}$ where every row/column pair $\{\boldsymbol{u}_i,\boldsymbol{v}_j^{\top}\}$ of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively are independent, we have shown in [1], that the posterior mean $\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^{-}$ and precision $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^{-},t+1}$ have the following expressions:

$$\widehat{\boldsymbol{Z}}_{\mathsf{p},t+1}^{-} = \widehat{\boldsymbol{U}}_{\mathsf{e},t+1}^{-}\,\widehat{\boldsymbol{V}}_{\mathsf{e},t+1}^{-\top} + \frac{\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{+},t}}{\beta}\,\widehat{\boldsymbol{Z}}_{\mathsf{e},t}^{+}\,\mathrm{Tr}\Big(\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^{-},t+1}\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^{-},t+1}\Big), \tag{4.34a}$$

$$\gamma_{\boldsymbol{Z}_{\mathsf{p}}^{-},t+1} = \gamma_{\boldsymbol{Z}_{\mathsf{e}}^{+},t} + MN\,\mathrm{Tr}\bigg(\frac{M\,N}{\beta}\,\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^{-},t+1}\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^{-},t+1} + N\,\boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^{-},t+1}\,\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^{-\top}\,\widehat{\boldsymbol{V}}_{\mathsf{p},t+1}^{-} + M\,\boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^{-},t+1}\,\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^{-\top}\,\widehat{\boldsymbol{U}}_{\mathsf{p},t+1}^{-}\bigg)^{-1}, \tag{4.34b}$$

which correspond to lines 31–32 in Algorithm 3. The extrinsic values, $\widehat{z}_{ij,\mathsf{e},t+1}^{-}$ and $\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{-},t+1}^{-1}$, for message (5') in Fig. 4.5 are then easily evaluated. Finally, the extrinsic mean and variance, $\widehat{z}_{ij,\mathsf{e},t+1}^{+}$ and $\gamma_{\boldsymbol{Z}_{\mathsf{e}}^{+},t+1}^{-1}$, of message (6') can be calculated from the posterior mean, $\widehat{z}_{ij,\mathsf{p},t+1}^{+}$ and common precision, $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^{+},t+1}$, as specified in lines 36–37 of Algorithm 3.

We refer the reader to the BiG-VAMP paper [1] for further details on this derivation. To tailor the non-linear denoiser in (4.31) and (4.32), we first restrict the prior on each entry $\widetilde{y}_{ij}$ of $\widetilde{\boldsymbol{Y}}$ as a quantization of each entry z_{ij} of $\boldsymbol{Z}$ over the

discrete set $\boldsymbol{e}$ to model the discrete support of the observed matrix. Moreover, for a fixed realization s_{ij} of $\mathsf{s}_{i,j}$, we set the selection channel $p_{\mathsf{y}_{ij}|\mathsf{z}_{ij}}(y_{ij}|z_{ij})$ corresponding to the transformation $y_{ij} = \phi_s(\phi_q(z_{ij}), \rho_0)$ to the data selection channel given explicitly by:

$$p_{\mathsf{y}_{ij}|\mathsf{z}_{ij}}(y_{ij}|z_{ij}) = s_{ij}\,\delta(y_{ij} - \phi_q(z_{ij})) + (1 - s_{ij})\,\delta(y_{ij}). \tag{4.35}$$

then by plugging (4.35) into (4.31) and (4.32), we compute in (4.36), (4.39a), and (4.37), (4.39b), the posterior mean $\widehat{z}^{+}_{ij,\mathsf{p}}$ and the posterior variance $\gamma^{-1}_{\mathsf{z}^{+}_{ij,\mathsf{p}}}$ for both cases when $s_{ij} = 1$ and $s_{ij} = 0$, respectively. In the case of an observed entry (e.g., $s_{ij} = 1$), we have $y_{ij} = \phi_q(z_{ij})$ is satisfied when $z_{ij} \in\,]r_t, r_{t+1}]$, so the posterior mean and variance will be computed as follows:

$$\widehat{z}^{+}_{ij,\mathsf{p}} = \frac{\int_{r_t}^{r_{t+1}} z_{ij}\,\mathcal{N}(z_{ij}; \widehat{z}^{-}_{ij,\mathsf{e}}, \gamma^{-1}_{\mathsf{z}^{-}_{ij,\mathsf{e}}})\,dz_{ij}}{\int_{r_t}^{r_{t+1}} \mathcal{N}(z_{ij}; \widehat{z}^{-}_{ij,\mathsf{e}}, \gamma^{-1}_{\mathsf{z}^{-}_{ij,\mathsf{e}}})\,dz_{ij}} = \widehat{z}^{-}_{ij,\mathsf{e}} - \frac{\sqrt{\gamma^{-1}_{\mathsf{z}^{-}_{ij,\mathsf{e}}}}}{\sqrt{2\pi}} \frac{e^{-\frac{\eta_2{}^2}{2}} - e^{-\frac{\eta_1{}^2}{2}}}{\Phi(\eta_2) - \Phi(\eta_1)}, \tag{4.36}$$

$$\gamma^{-1}_{\mathsf{z}^{+}_{ij,\mathsf{p}}} = \gamma^{-1}_{\mathsf{z}^{-}_{ij,\mathsf{e}}} \left(1 - \frac{1}{\sqrt{2\pi}} \left[\frac{\eta_2 e^{-\frac{\eta_2{}^2}{2}} - \eta_1 e^{-\frac{\eta_1{}^2}{2}}}{\Phi(\eta_2) - \Phi(\eta_1)} + \frac{1}{\sqrt{2\pi}} \left(\frac{e^{-\frac{\eta_2{}^2}{2}} - e^{-\frac{\eta_1{}^2}{2}}}{\Phi(\eta_2) - \Phi(\eta_1)}\right)^2\right]\right), \tag{4.37}$$

where

$$\eta_1 = \sqrt{\gamma_{\mathsf{z}^{-}_{ij,\mathsf{e}}}}\,(r_t - \widehat{z}^{-}_{ij,\mathsf{e}}), \tag{4.38a}$$

$$\eta_2 = \sqrt{\gamma_{\mathsf{z}^{-}_{ij,\mathsf{e}}}}\,(r_{t+1} - \widehat{z}^{-}_{ij,\mathsf{e}}). \tag{4.38b}$$

And in the case where $y_{ij} = 0$ (e.g., $s_{ij} = 0$), the posterior mean and variance remain the same as shown below:

$$\widehat{z}^{+}_{ij,\mathsf{p}} = \widehat{z}^{-}_{ij,\mathsf{e}}, \tag{4.39a}$$

$$\gamma^{-1}_{\mathsf{z}^{+}_{ij,\mathsf{p}}} = \gamma^{-1}_{\mathsf{z}^{-}_{ij,\mathsf{e}}}. \tag{4.39b}$$

Finally, we implement this quantized denoiser upon the BiG-VAMP algorithmic steps along with the iterative scheduling between the four steps, described above, to obtain our proposed BiG-VAMP-MC shown in Algorithm 3.

4.2.4 Thresholds learning using EM algorithm

In some real-world applications (e.g., A movie or any type of RSs ...), the fully low-rank quantization dense matrix $\widetilde{\boldsymbol{Y}}$ is unknown, unlike in the synthetic

case where we create it and know it beforehand. Therefore, learning the thresholds $[r_1, r_2, \ldots, r_{l-1}]$ and updating them at every iteration allows the algorithm to estimate $\boldsymbol{Y}$ more accurately. This iterative thresholds learning process aims to find the maximum likelihood (ML) estimate $\widehat{\boldsymbol{r}}$ of $\boldsymbol{r} \triangleq [r_1, r_2, \ldots, r_{l-1}]^\top$ over all the observed data of the matrix $\boldsymbol{Y}$ as follows:

$$\widehat{\boldsymbol{r}} \triangleq \underset{\boldsymbol{r}}{\operatorname{argmax}} \ \log \left[\prod_{i=1}^{N} \prod_{j=1}^{M} p_{\mathsf{y}_{ij}}(y_{ij}; \boldsymbol{r}) \right], \tag{4.40}$$

where $p_{\mathsf{y}_{ij}}(y_{ij}; \boldsymbol{r})$ is the pdf of y_{ij} paramatrized by $\boldsymbol{r}$ given by:

$$p_{\mathsf{y}_{ij}}(y_{ij}; \boldsymbol{r}) \propto \int_{-\infty}^{+\infty} \mathcal{N}(z_{ij}; \widehat{z_{ij,\mathsf{e}}^{-}}, \beta^{-1}\gamma_{z_{ij,\mathsf{e}}^{-}}^{-1} \boldsymbol{I}) \ p_{\mathsf{y}_{ij}|\mathsf{z}_{ij}}(y_{ij}|z_{ij}; \boldsymbol{r}) \ dz_{ij}. \tag{4.41}$$

So the total log-likelihood or the objective function to be maximized will be:

$$f(\boldsymbol{r}, \boldsymbol{Y}, \widehat{\boldsymbol{Z}}_{\mathsf{e}}^{-}) \triangleq \sum_{i=1}^{N} \sum_{j=1}^{M} f_{ij}(\boldsymbol{r}, y_{ij}, \widehat{z_{ij,\mathsf{e}}^{-}}), \tag{4.42}$$

where the ij-th objective function component $f_{ij}(\cdot)$ is defined as follows:

$$f_{ij}(\boldsymbol{r} \triangleq [r_1, r_2, \ldots, r_{l-1}]^\top, y_{ij}, \widehat{z_{ij,\mathsf{e}}^{-}}) \triangleq$$
$$\log \left[\Phi\left(\sqrt{\gamma_{\widehat{\mathsf{z}}_{ij,e}^{-}}} \left(r^{\text{upper}} \ (y_{ij}) - \widehat{z_{ij,\mathsf{e}}^{-}} \right) \right) - \ \Phi\left(\sqrt{\gamma_{\widehat{\mathsf{z}}_{ij,e}^{-}}} \left(r^{\text{lower}} \ (y_{ij}) - \widehat{z_{ij,\mathsf{e}}^{-}} \right) \right) \right] (1 - \delta(y_{ij})), \tag{4.43}$$

with $r^{\text{lower}}(y_{ij})$ and $r^{\text{upper}}(y_{ij})$ refer to the two consecutive thresholds where:

$$\widehat{z_{ij,\mathsf{e}}^{-}} \ \in \ \rfloor \ r^{\text{lower}} \ (y_{ij}) \ , \ r^{\text{upper}} \ (y_{ij}) \ \lfloor .$$

The objective function in (4.42) is a nonlinear transformation of the parameters, so solving the maximization problem (4.40) analytically is mathematically intractable. However, we can solve this optimization problem numerically by using iterative methods. In this paper, we use the EM approach [40], which is a commonly used tool for vector parameter estimation. Interestingly, the BiG-VAMP method provides the necessary posterior probabilities required by the EM algorithm to update its estimates accurately. To successfully use the EM algorithm, it's important to identify the incomplete and complete data sets that are relevant to the estimation problem. In our case, the incomplete data set is represented by $\boldsymbol{Y}$, while the complete data set is represented by $\boldsymbol{H} \triangleq [\boldsymbol{Y}, \boldsymbol{Z}] \in \mathbb{R}^{2N \times M}$. Instead of maximizing the actual log-likelihood function (LLF), which is $\log [p_{\boldsymbol{\mathsf{Y}}}(\mathbf{Y}; \boldsymbol{r})]$, the

EM algorithm maximizes $\log\left[p_{\mathbf{H}}(\boldsymbol{H};\boldsymbol{r})\right]$. The complete data set, $\boldsymbol{H}$, is not entirely available, so the EM algorithm uses the conditional expectation of $\log\left[p_{\mathbf{H}}(\boldsymbol{H};\boldsymbol{r})\right]$ instead.

$$\mathbb{E}_{\mathbf{H}|\mathbf{Y}}\Big\{\log\left[p_{\mathbf{H}}(\boldsymbol{H};\boldsymbol{r})\right] \mid \boldsymbol{Y}\Big\} = \int_{\boldsymbol{H}} \log\left[p_{\mathbf{Y}}(\mathbf{Y};\boldsymbol{r})\right] p_{\mathbf{H}|\mathbf{Y}}(\boldsymbol{H} \mid \boldsymbol{Y};\boldsymbol{r})\, \mathrm{d}\boldsymbol{H}. \tag{4.44}$$

Now, using the fact that:

$$p_{\mathbf{H}|\mathbf{Y}}(\boldsymbol{H} \mid \boldsymbol{Y};\boldsymbol{r}) = p_{\mathbf{Z}|\mathbf{Y}}(\boldsymbol{Z} \mid \boldsymbol{Y};\boldsymbol{r}), \tag{4.45}$$

and by plugging (4.45) back in (4.44), the expected LLF in (4.44) will have the following expression:

$$\mathbb{E}_{\mathbf{Z}|\mathbf{Y}}\Big\{\log\left[p_{\mathbf{Y}}(\boldsymbol{Y};\boldsymbol{r})\right] \mid \boldsymbol{Y}\Big\} = \int_{\boldsymbol{Z}} \log\left[p_{\mathbf{Y}}(\mathbf{Y};\boldsymbol{r})\right] p_{\mathbf{Z}|\mathbf{Y}}(\boldsymbol{Z} \mid \boldsymbol{Y};\boldsymbol{r})\, \mathrm{d}\boldsymbol{Z}. \tag{4.46}$$

The EM algorithm is an iterative algorithm that starts with an initial guess, denoted as $\widehat{\boldsymbol{r}}^{\,(0)}$, for the unknown parameter vector $\boldsymbol{r}$. At each iteration, the algorithm computes a new estimate, $\widehat{\boldsymbol{r}}_k$, of the maximum likelihood estimate (MLE) of $\boldsymbol{r}$. The EM algorithm alternates between two main steps: the E-step and the M-step.

In the E-step, the algorithm computes the auxiliary function, $Q\left(\boldsymbol{r}, \widehat{\boldsymbol{r}}_k\right)$, which is the expected value of the log-likelihood function with respect to the conditional distribution of the missing data given the observed data and the current k^{th} estimate of $\boldsymbol{r}$; $\widehat{\boldsymbol{r}}_k$:

$$Q\left(\boldsymbol{r}; \widehat{\boldsymbol{r}}_k\right) = \mathbb{E}_{\mathbf{Z}|\mathbf{Y};\widehat{\boldsymbol{r}}_k}\Big\{\log\left[p_{\mathbf{Y}}\left(\boldsymbol{Y};\boldsymbol{r}\right)\right]\Big\}. \tag{4.47}$$

or the elements of the observed matrix $\boldsymbol{Y}$ are i.i.d.:

$$p_{\mathbf{Y}}\left(\boldsymbol{Y};\boldsymbol{r}\right) = \prod_{i=1}^{N}\prod_{j=1}^{M} p_{\mathsf{y}_{ij}}(y_{ij};\boldsymbol{r}). \tag{4.48}$$

Hence, we rewrite (4.47) as follows:

$$Q\left(\boldsymbol{r}; \widehat{\boldsymbol{r}}_k\right) \propto \sum_{i=1}^{N}\sum_{j=1}^{M} \mathbb{E}_{\mathbf{Z}|\mathbf{Y};\widehat{\boldsymbol{r}}_k}\Big\{\log\left[p_{\mathsf{y}_{ij}}(y_{ij};\boldsymbol{r})\right]\Big\}. \tag{4.49}$$

Recall that the expectation in (4.49) is taken with respect to the posterior density

$p_{\mathbf{Z}|\mathbf{Y}}\left(\boldsymbol{Z} \mid \boldsymbol{Y}; \widehat{\boldsymbol{r}}_k\right)$ which we further approximate by:

$$p_{\mathbf{Z}|\mathbf{Y}}\left(\boldsymbol{Z} \mid \boldsymbol{Y}; \widehat{\boldsymbol{r}}_k\right) = \prod_{i=1}^{N}\prod_{j=1}^{M} p_{\mathsf{z}_{ij}|\mathsf{y}_{ij}}\left(z_{ij} \mid y_{ij}; \boldsymbol{r}\right), \tag{4.50}$$

where for a fixed realization z_{ij} of $\widehat{\mathsf{z}}_{ij,\mathsf{e}}$, the conditional probability distribution $p_{\mathsf{z}_{ij}|\mathsf{y}_{ij}}\left(z_{ij} \mid y_{ij}; \boldsymbol{r}\right)$ corresponding to the transformation $y_{ij} = \phi_s(\phi_q(z_{ij}), \rho_0)$ is explicitly given by:

$$p_{\mathsf{z}_{ij}|\mathsf{y}_{ij}}\left(z_{ij} \mid y_{ij}; \boldsymbol{r}\right) \propto \mathcal{N}(z_{ij}; \widehat{z}_{ij,\mathsf{e}}, \beta^{-1}\gamma_{z_{ij,\mathsf{e}}}^{-1}\boldsymbol{I})\; p_{\mathsf{y}_{ij}|\mathsf{z}_{ij}}(y_{ij}|z_{ij}; \boldsymbol{r}). \tag{4.51}$$

Using (4.51) and (4.50) in (4.49), it follows that:

$$\begin{aligned} Q\left(\boldsymbol{r}; \widehat{\boldsymbol{r}}_k\right) &\propto \sum_{i=1}^{N}\sum_{j=1}^{M} \mathbb{E}_{\mathsf{z}|\mathsf{y}_{ij};\widehat{\boldsymbol{r}}_k}\Big\{p_{\mathsf{y}_{ij}}(y_{ij}; \boldsymbol{r})\Big\}, \\ &\propto \sum_{i=1}^{N}\sum_{j=1}^{M} \int_z f_{ij}(\boldsymbol{r}, y_{ij}, \widehat{z}_{ij,\mathsf{e}})\; p_{\mathsf{z}|\mathsf{y}_{ij};\widehat{\boldsymbol{r}}_k}\left(z \mid y_{ij}; \widehat{\boldsymbol{r}}_k\right) dz. \end{aligned} \tag{4.52}$$

In the M-step, the algorithm maximizes the auxiliary function, $Q\left(\boldsymbol{r}; \widehat{\boldsymbol{r}}_k\right)$ with respect to the parameters of the model $\boldsymbol{r}$:

$$\widehat{\boldsymbol{r}}_{k+1} = \underset{\boldsymbol{r}}{\operatorname{argmax}}\; Q\left(\boldsymbol{r}; \widehat{\boldsymbol{r}}_k\right). \tag{4.53}$$

Then we perform the gradient descent steps to solve the maximization problem in (4.53) and update the old model $\widehat{\boldsymbol{r}}_k$ to the new one $\widehat{\boldsymbol{r}}_{k+1}$, and return to the E-Step, then keep iterating these steps till convergence. The algorithm stops when either the user-specified convergence criterion $\left\|\widehat{\boldsymbol{r}}_{k+1} - \widehat{\boldsymbol{r}}_k\right\|_{\mathrm{F}} \leq \epsilon_{\,\mathsf{EM}}$ is met or a predefined maximum number of iterations $k_{\,\mathsf{EM}}$ is reached.

4.3 State evolution

The goal of this section is to understand the asymptotic regime of the BiG-VAMP-MC algorithm (i.e., as $N, M \to \infty$) for the i.i.d. class of matrices. The corresponding asymptotic regime refers to the scenario where $R, N, M \to +\infty$ with $\frac{R}{N} \to \beta_u = \mathcal{O}(1)$ and $\frac{R}{M} \to \beta_v = \mathcal{O}(1)$ for some fixed ratios $\beta_u \leq 1$ and $\beta_v \leq 1$. Result 1 and Result 2 summarize the main analytical expressions we derived for the customized MMSE and LMMSE estimators of $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{Z}$, respectively for BiG-VAMP-MC. See [1] for the full derivation. For large enough M and N, we

use of the approximation:

$$\langle \boldsymbol{Y} \odot \boldsymbol{Y} \rangle \;\approx\; \bar{\sigma}_u^2 \bar{\sigma}_v^2 r \,+\, \sqrt{MN}\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+}^{-1}, \tag{4.54}$$

which follows from the observation that $\sigma_u^2 \triangleq \langle \boldsymbol{U} \odot \boldsymbol{U} \rangle \approx \bar{\sigma}_u^2 \triangleq \mathbb{E}[\mathsf{u}_{i,\ell}^2 | p_{\mathsf{u}}(u)]$ and $\sigma_v^2 \triangleq \langle \boldsymbol{V} \odot \boldsymbol{V} \rangle \approx \bar{\sigma}_v^2 \triangleq \mathbb{E}[\mathsf{v}_{j\ell}^2 | p_{\mathsf{v}}(v)]$ $\forall i, j, \ell$. Here, $p_{\mathsf{u}}(u)$ [resp., $p_{\mathsf{v}}(v)$] is a common prior on the entries of the matrix $\mathbf{U}$ [resp., $\mathbf{V}$]. In AMP practices, a SE ansatz is based on the following concentration of measure for the intrinsic/extrinsic and in/out precision variables in the asymptotic regime, i.e.:

$$\lim_{M,N\to\infty} \left(\gamma_{\boldsymbol{U}_{\mathsf{p}}^+,t}, \gamma_{\boldsymbol{U}_{\mathsf{e}}^+,t}\right) = \left(\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^+,t}, \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,t}\right), \tag{4.55a}$$

$$\lim_{M,N\to\infty} \left(\gamma_{\boldsymbol{U}_{\mathsf{p}}^-,t}, \gamma_{\boldsymbol{U}_{\mathsf{e}}^-,t}\right) = \left(\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-,t}, \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-,t}\right), \tag{4.55b}$$

$$\lim_{M,N\to\infty} \left(\gamma_{\boldsymbol{V}_{\mathsf{p}}^+,t}, \gamma_{\boldsymbol{V}_{\mathsf{e}}^+,t}\right) = \left(\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^+,t}, \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,t}\right), \tag{4.55c}$$

$$\lim_{M,N\to\infty} \left(\gamma_{\boldsymbol{V}_{\mathsf{p}}^-,t}, \gamma_{\boldsymbol{V}_{\mathsf{e}}^-,t}\right) = \left(\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-,t}, \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-,t}\right), \tag{4.55d}$$

$$\lim_{M,N\to\infty} \left(\gamma_{\boldsymbol{Z}_{\mathsf{p}}^+,t}, \gamma_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\right) = \left(\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^+,t}, \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\right), \tag{4.55e}$$

$$\lim_{M,N\to\infty} \left(\gamma_{\boldsymbol{Z}_{\mathsf{p}}^-,t}, \gamma_{\boldsymbol{Z}_{\mathsf{e}}^-,t}\right) = \left(\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^-,t}, \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-,t}\right). \tag{4.55f}$$

Result 1 *The expressions of the customized MMSE estimators of $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{Z}$, given the threshold vector $\boldsymbol{r} \triangleq [r_1, r_2, \ldots, r_{l-1}]^\top$ are:*

$$\mathcal{E}_{u^+}(\gamma_{\boldsymbol{U}_e^-}) \;\triangleq\; \frac{1}{\gamma_{\boldsymbol{U}_p^+}} \;=\; \frac{1}{N\,\gamma_{\boldsymbol{U}_e^-}} \sum_{i=1}^{N} \langle \mathbf{g}'_{\mathsf{u}}(\widehat{\boldsymbol{u}}_i^-, \gamma_{\boldsymbol{U}_e^-}^{-1}) \rangle, \tag{4.56}$$

$$\mathcal{E}_{v^+}(\gamma_{\boldsymbol{V}_e^-}) \;\triangleq\; \frac{1}{\gamma_{\boldsymbol{V}_p^+}} \;=\; \frac{1}{M\,\gamma_{\boldsymbol{V}_e^-}} \sum_{j=1}^{M} \langle \mathbf{g}'_{\mathsf{v}}(\widehat{\boldsymbol{v}}_j^-, \gamma_{\boldsymbol{U}_e^-}^{-1}) \rangle, \tag{4.57}$$

$$\mathcal{E}_{z^+}(\gamma_{\boldsymbol{Z}_e^-}, \boldsymbol{r}) \;\triangleq\; \frac{1}{\gamma_{\boldsymbol{Z}_p^+}} = \frac{1}{MN\,\gamma_{\boldsymbol{Z}_e^-}} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{g}'_{\mathsf{z}}(y_{ij}, \widehat{z}_{ij,\mathsf{e}}^-, \gamma_{\boldsymbol{Z}_e^-}^{-1}, \boldsymbol{r}). \tag{4.58}$$

where the quantization denoising function $\mathbf{g}_{\mathsf{z}}(y_{ij}, \widehat{z}_{ij,\mathsf{e}}^-, ., \boldsymbol{r})$ in (4.58), is computed throughout the steps described in Section 4.1.3 to obtain the theoretical posterior variance $\gamma_{z_{ij,p}^+}^{-1}$.

In the large system limits, the empirical averages involved in (4.56),(4.57) and (4.58) can be approximated by the following statistical average in which we use the

fact that $\gamma_{\boldsymbol{U}_{\mathsf{p}}^+} \longrightarrow \bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^+}$, $\gamma_{\boldsymbol{V}_{\mathsf{p}}^+} \longrightarrow \bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^+}$ *and* $\gamma_{\boldsymbol{Z}_{\mathsf{p}}^+} \longrightarrow \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^+}$ *as* M *and* N *grow large:*

$$\mathcal{E}_{u^+}(\bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-}) \triangleq \frac{1}{\bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-}} \mathbb{E}\left[g'_{\mathsf{u}}(\widehat{u}_{\mathsf{e}}^-, \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-}^{-1}) \middle| p_{\mathsf{u}}(u) p_{\widehat{\mathsf{u}}_{\mathsf{e}}^-|\mathsf{u}}(\widehat{u}_{\mathsf{e}}^-|u) \right], \tag{4.59}$$

$$\mathcal{E}_{v^+}(\bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-}) \triangleq \frac{1}{\bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-}} \mathbb{E}\left[g'_{\mathsf{v}}(\widehat{v}_{\mathsf{e}}^-, \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-}^{-1}) \middle| p_{\mathsf{v}}(v) p_{\widehat{\mathsf{v}}_{\mathsf{e}}^-|\mathsf{v}}(\widehat{v}_{\mathsf{e}}^-|v) \right], \tag{4.60}$$

$$\mathcal{E}_{z^+}(\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-}, \boldsymbol{r}) \triangleq \frac{1}{\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-}} \mathbb{E}\left[g'_{\mathsf{z}}(y, \widehat{z}_{\mathsf{e}}^-, \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-}^{-1}, \boldsymbol{r}) \middle| p_{\mathsf{z}}(z) p_{y|\mathsf{z}}(y|z;\boldsymbol{r}) p_{\widehat{\mathsf{z}}_{\mathsf{e}}^-|\mathsf{z}}(\widehat{z}_{\mathsf{e}}^-|z) \right]. \tag{4.61}$$

In (4.59), (4.60) and (4.61), $p_{\widehat{\mathsf{u}}_{\mathsf{e}}^-|\mathsf{u}}(\widehat{u}_{\mathsf{e}}^-|u)$, $p_{\widehat{\mathsf{v}}_{\mathsf{e}}^-|\mathsf{v}}(\widehat{v}_{\mathsf{e}}^-|v)$ and $p_{\widehat{\mathsf{z}}_{\mathsf{e}}^-|\mathsf{z}}(\widehat{z}_{\mathsf{e}}^-|z)$ correspond to the scalar models $\widehat{\mathsf{u}}_{\mathsf{e}}^- = \mathsf{u} + \mathsf{w}_u$, $\widehat{\mathsf{v}}_{\mathsf{e}}^- = \mathsf{v} + \mathsf{w}_v$ and $\widehat{\mathsf{z}}_{\mathsf{e}}^- = \mathsf{z} + \mathsf{w}_z$, respectively, where under the matched conditions, we have $\mathsf{w}_u \sim \mathcal{N}(w_u; 0, \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-}^{-1})$, $\mathsf{w}_v \sim \mathcal{N}(w_v; 0, \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-}^{-1})$ and $\mathsf{w}_z \sim \mathcal{N}(w_z; 0, \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-}^{-1})$.

Result 2 *The expressions of the LMMSE estimators of* $\boldsymbol{U}$, $\boldsymbol{V}$ *and* $\boldsymbol{Z}$ *are:*

$$\mathcal{E}_{u^-}(\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+}) = \lim_{r\to\infty} \frac{1}{r} \operatorname{Tr}\left(\left[\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+} \boldsymbol{I}_r + \frac{\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+}}{\sqrt{MN}} \widehat{\boldsymbol{V}}_{\mathsf{p}}^{-\top} \widehat{\boldsymbol{V}}_{\mathsf{p}}^- \right]^{-1} \right), \tag{4.62}$$

$$\mathcal{E}_{v^-}(\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+}) = \lim_{r\to\infty} \frac{1}{r} \operatorname{Tr}\left(\left[\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+} \boldsymbol{I}_r + \frac{\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+}}{\sqrt{MN}} \widehat{\boldsymbol{U}}_{\mathsf{p}}^{-\top} \widehat{\boldsymbol{U}}_{\mathsf{p}}^- \right]^{-1} \right), \tag{4.63}$$

$$\mathcal{E}_{z^-}(\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+}) = \left(\gamma_{\boldsymbol{Z}_{\mathsf{e}}^+} + \sqrt{MN} \left[\operatorname{Tr}\left(\frac{1}{\beta} \boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-} \boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-}^{\top} + \frac{1}{M} \boldsymbol{R}_{\boldsymbol{U}_{\mathsf{p}}^-} \widehat{\boldsymbol{V}}_{\mathsf{p}}^{-\top} \widehat{\boldsymbol{V}}_{\mathsf{p}}^- + \frac{1}{N} \boldsymbol{R}_{\boldsymbol{V}_{\mathsf{p}}^-} \widehat{\boldsymbol{U}}_{\mathsf{p}}^{-\top} \widehat{\boldsymbol{U}}_{\mathsf{p}}^- \right) \right]^{-1} \right)^{-1}. \tag{4.64}$$

where

$$\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+} = \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+} + \frac{1}{\beta} \sqrt{\beta_u/\beta_v} \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+} \bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-}^{-1} - \sqrt{\beta_u/\beta_v} \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+} \bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-}^{-1} \left(\sqrt{\beta_u \beta_v} \bar{\sigma}_u^2 \bar{\sigma}_v^2 \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+} + 1 \right), \tag{4.65}$$

$$\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+} = \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+} + \frac{1}{\beta} \sqrt{\beta_v/\beta_u} \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+} \bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-}^{-1} - \sqrt{\beta_v/\beta_u} \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+} \bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-}^{-1} \left(\sqrt{\beta_u \beta_v} \bar{\sigma}_u^2 \bar{\sigma}_v^2 \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+} + 1 \right). \tag{4.66}$$

For large enough M *and* N *and given the concentration of measure for the intrinsic/extrinsic precisions variables defined in (4.55), the LMMSE estimators of* $\boldsymbol{U}$, $\boldsymbol{V}$ *and* $\boldsymbol{Z}$ *are given by* [2]*:*

[2]For a detailed derivation on how to use the asymptotic theorem on random matrix theorem mentioned in section 2.5 to approximate the LMMSE estimators in the large system limit, we refer the reader to check [1]. Specifically, the paper explains the derivation steps that transform the equations (4.62) into (4.67) and (4.63) into (4.68).

$$\mathcal{E}_{u^-}(\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+}) \;= \widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+}^{-1}\left(1-\frac{\mathcal{F}(\alpha_v,\beta_v)}{4\beta_v\alpha_v}\right), \tag{4.67}$$

$$\mathcal{E}_{v^-}(\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+}) \;= \widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+}^{-1}\left(1-\frac{\mathcal{F}(\alpha_u,\beta_u)}{4\beta_u\alpha_u}\right), \tag{4.68}$$

$$\mathcal{E}_{z^-}(\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+}) \;= \left(\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+}+\frac{1}{\sqrt{\beta_u\beta_v}}\Big[\frac{1}{\beta}\;\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-}^{-1}\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-}^{-1}+\frac{(\bar{\sigma}_v^2-\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-}^{-1})}{\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-}}+\frac{(\bar{\sigma}_u^2-\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-}^{-1})}{\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-}}\Big]^{-1}\right)^{-1}. \tag{4.69}$$

where

$$\alpha_u \triangleq \sqrt{\beta_v/\beta_u}\;\frac{\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+}(\bar{\sigma}_u^2-\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-}^{-1})}{\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+}},\;\; \alpha_v \triangleq \sqrt{\beta_u/\beta_v}\;\frac{\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+}(\bar{\sigma}_v^2-\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-}^{-1})}{\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+}},$$

and

$$\mathcal{F}(x,z) \;\triangleq\; \left(\sqrt{x(1+\sqrt{z})^2+1}-\sqrt{x(1-\sqrt{z})^2+1}\;\right)^2,$$

Under all the assumptions, we can now describe in Algorithm 4 our main result, which is the SE recursion for BiG-VAMP-MC.

Conclusion

In this chapter, inspired by the concepts presented in the previous two chapters, and the low-rank matrix reconstruction technique, we introduce our primary solution for addressing discrete MF/MC problems. Our approach is based on the BiG-VAMP approach, its factor graph, and its derivation using BP and graphical modeling techniques

Algorithm 4 BiG-VAMP-MC State Evolution

Initialization : Temperature parameter β; extrinsic precisions as defined above extrinsic precisions $\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,1}$, $\bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,1}$, $\bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,1}$, $\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-,1}$, $\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-,1}$.

1: **for** $t = 1, \ldots, T_{\max}$ **do**

▷ compute the effective inverse noise variance for the Bi-LMMSE block

2: $$\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,t+1} = \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,t} + \tfrac{1}{\beta}\sqrt{\tfrac{\beta_u}{\beta_v}}\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\bar{\gamma}^{-1}_{\boldsymbol{V}_{\mathsf{p}}^-,t} - \sqrt{\tfrac{\beta_u}{\beta_v}}\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\bar{\gamma}^{-1}_{\boldsymbol{V}_{\mathsf{p}}^-,t}\left(\sqrt{\beta_u\beta_v}\bar{\sigma}_u^2\bar{\sigma}_v^2\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t} + 1\right)$$

3: $$\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,t+1} = \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,t} + \tfrac{1}{\beta}\sqrt{\tfrac{\beta_v}{\beta_u}}\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\bar{\gamma}^{-1}_{\boldsymbol{U}_{\mathsf{p}}^-,t} - \sqrt{\tfrac{\beta_v}{\beta_u}}\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\bar{\gamma}^{-1}_{\boldsymbol{U}_{\mathsf{p}}^-,t}\left(\sqrt{\beta_u\beta_v}\bar{\sigma}_u^2\bar{\sigma}_v^2\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t} + 1\right)$$

▷ compute the analytical posterior and extrinsic precision of $\mathbf{u}^-$

4: $$\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1} = \frac{1}{\mathcal{E}_{u^-}\left(\widetilde{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,t+1}\right)},\ \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-,t+1} = \bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^-,t+1} - \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,t}$$

▷ compute the analytical posterior and extrinsic precision of $\mathbf{v}^-$

5: $$\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1} = \frac{1}{\mathcal{E}_{v^-}\left(\widetilde{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,t+1}\right)},\bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-,t+1} = \bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^-,t+1} - \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,t}$$

▷ compute the analytical posterior and extrinsic precision of $\mathbf{z}^-$

6: $$\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^-,t+1} = \frac{1}{\mathcal{E}_{z^-}\left(\widetilde{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}\right)},\quad \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1} = \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^-,t+1} - \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t}$$

▷ compute the analytical posterior and extrinsic precision of $\mathbf{u}^+$

7: $$\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^+,t+1} = \frac{1}{\mathcal{E}_{u^+}(\bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^-,t+1})},\quad \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e}}^+,t+1} = \bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^+,t+1} - \bar{\gamma}_{\boldsymbol{U}_{\mathsf{e},t+1}^-}$$

▷ compute the analytical posterior and extrinsic precision of $\mathbf{v}^+$

8: $$\bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^+,t+1} = \frac{1}{\mathcal{E}_{v^+}(\bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-,t+1})},\quad \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^+,t+1} = \bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^+,t+1} - \bar{\gamma}_{\boldsymbol{V}_{\mathsf{e}}^-,t+1}$$

▷ compute the analytical posterior and extrinsic precision of $\mathbf{z}^+$

9: $$\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^+,t+1} = \frac{1}{\mathcal{E}_{z^+}(\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1},\boldsymbol{r})},\bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^+,t+1} = \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^+,t+1} - \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{e}}^-,t+1}$$

10: **end for**

11: **Return** $\bar{\gamma}_{\boldsymbol{U}_{\mathsf{p}}^+,T_{\max}+1}, \bar{\gamma}_{\boldsymbol{V}_{\mathsf{p}}^+,T_{\max}+1}, \bar{\gamma}_{\boldsymbol{Z}_{\mathsf{p}}^+,T_{\max}+1}$

Chapter 5

Simulation Results & Discussions

Introduction

In this chapter, we perform different experiments on synthetic and real-world data. Synthetic experiments are conducted to evaluate the algorithm's characteristics in controlled environments with specific known data structures, which demonstrate what kind of data each algorithm excels in and what their limitations are, while real-world experiments are aimed at comparing the performance of the algorithms with actual data.

5.1 Evaluation Metrics

We run all AMP-based algorithms with a maximum number of iterations $T_{\max} = 1000$ and $N_{\mathrm{MC}} = 100$ Monte-Carlo trials for different values. For the experiments, the assessment of the deviation of the estimated discrete matrix $\widehat{\boldsymbol{Y}}$ from the true discrete matrix $\boldsymbol{Y}$, both of size $N \times M$, is measured via the following distance metrics:

- *Normalized mean square error* (NMSE):

$$\mathrm{NMSE} \triangleq \frac{1}{N_{\mathrm{MC}}} \sum_{\ell=1}^{N_{\mathrm{MC}}} \frac{\left\| \boldsymbol{Y}_\ell - \widehat{\boldsymbol{Y}}_\ell \right\|_F^2}{\|\boldsymbol{Y}_\ell\|_F^2}.$$

- *Projected normalized mean square error* (P-NMSE):

$$\text{P-NMSE} \triangleq \frac{1}{N_{\mathrm{MC}}} \sum_{\ell=1}^{N_{\mathrm{MC}}} \frac{\left\| P_{\Omega^c}(\boldsymbol{Y}_\ell - \widehat{\boldsymbol{Y}}_\ell) \right\|_F^2}{\|P_{\Omega^c}(\boldsymbol{Y}_\ell)\|_F^2}.$$

where $P_\Omega(\cdot)$ indicates the mask operator (i.e., projection) defined for any matrix $\boldsymbol{A}$ as follows:

$$[P_\Omega(\boldsymbol{A})]_{ij} = \begin{cases} [\boldsymbol{A}]_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases}$$

Here, Ω denotes the observed index set with Ω^c being its complementary set. This metric is particularly used to evaluate discreteness-aware methods only the unobserved data is considered. We also define the sparsity and the observed ratios as, $\rho \triangleq \frac{|O|}{NM}$, $\delta \triangleq \frac{|\Omega|}{(NM-|O|)}$, respectively, where $|\Omega|$ denotes the number of elements in the observed index set and $|O|$ denotes the number of missing entries of the matrix $\boldsymbol{Y}$.

5.2 Baseline MF methods

We visualize the performance of BiG-VAMP-MC against state-of-the-art techniques, namely:

- Quantized bilinear generalized AMP for discrete MC (BiG-AMP-MC) [37] which we customized to incorporate the discreteness on $\boldsymbol{Y}$ and on the latent factor matrix $\boldsymbol{U}$,
- Nonconvex inexact accelerated proximal gradient (niAPG) algorithm for nuclear norm regularized least squares problems method (APG) developed in [39],
- Discrete-aware MC via efficient inexact proximal gradient algorithm (DA-niAPG) proposed in [21].

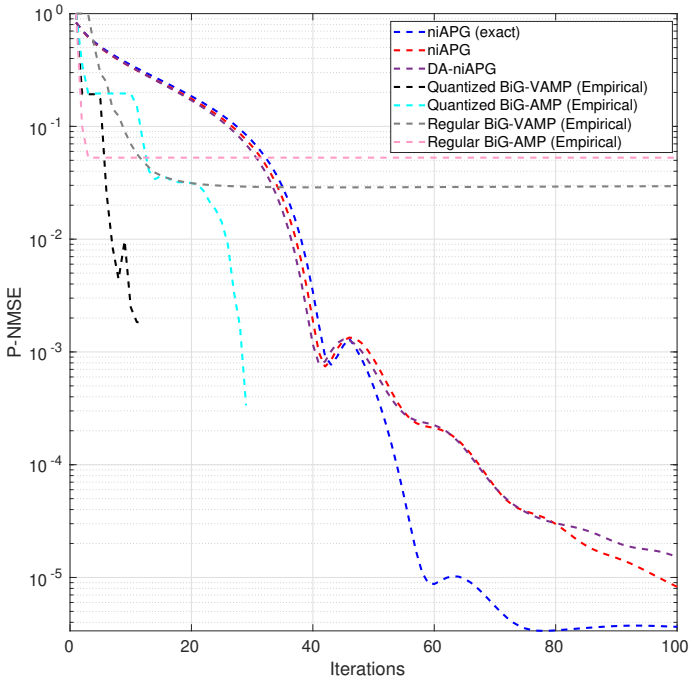
5.3 Experiments

As per BiG-VAMP initialization, we set the temperature parameter β and the convergence precision tolerance ξ to 1 (i.e., MMSE case) and 10^{-6}, respectively. The results exposed in the sequel show that the BiG-VAMP-MC produces the best reconstruction performance.

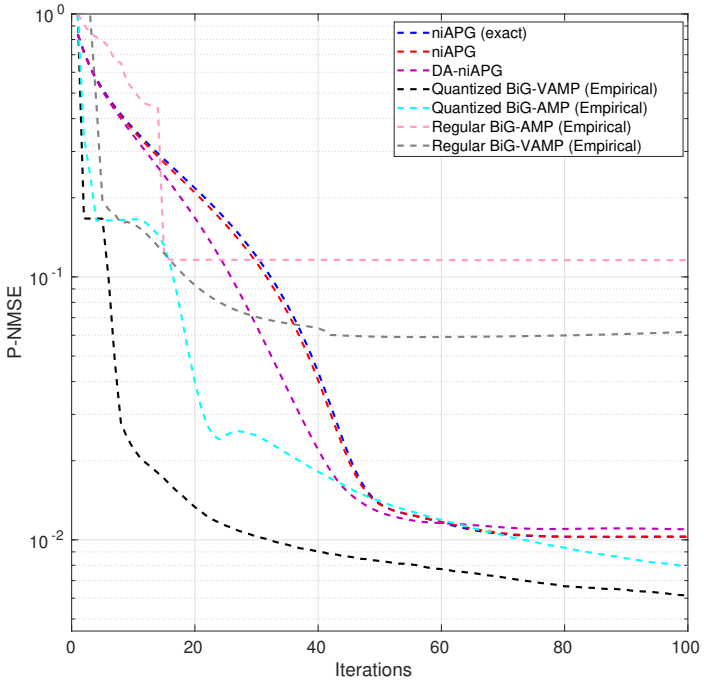
5.3.1 Experiments on synthetic data

First, in Fig. 5.1, we compared BiG-VAMP-MC to the state-of-art algorithms for the discrete-aware MC problem on a discrete matrix, $\boldsymbol{Y}$, with $N = 400$, $M = 300$,

and $R = 2$, synthetically generated from the following 5-symbol and threshold

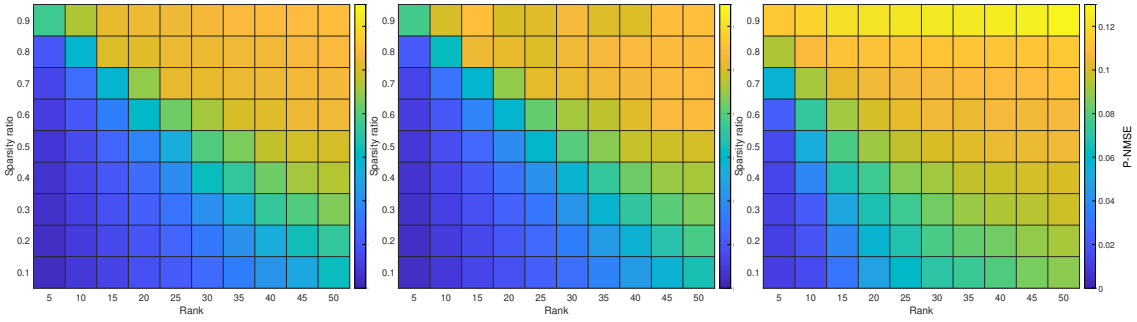


(a) $\boldsymbol{U}$ being binary and $\boldsymbol{V}$ being Gaussian.



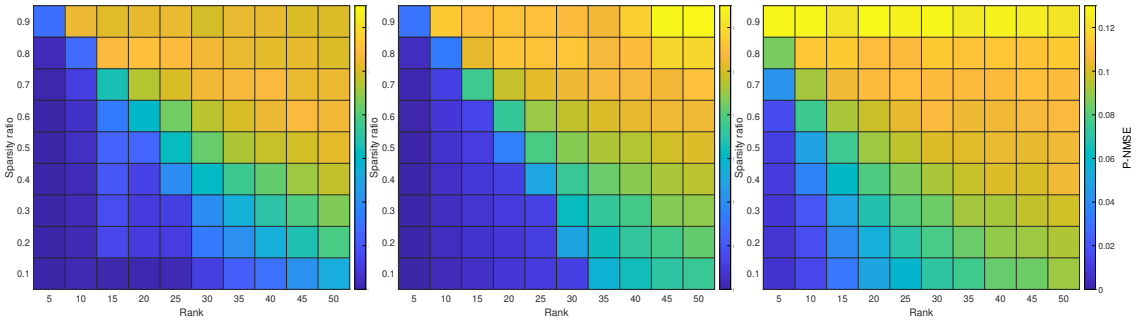
(b) $\boldsymbol{U}$ and $\boldsymbol{V}$ being Gaussian.

Figure 5.1: P-NMSE of the proposed solution as compared to the state-of-the-art algorithms for the discrete-aware MC problem on a synthetically generated discrete matrix, Y, with $N = 400$, $M = 300$ and $R = 2$.



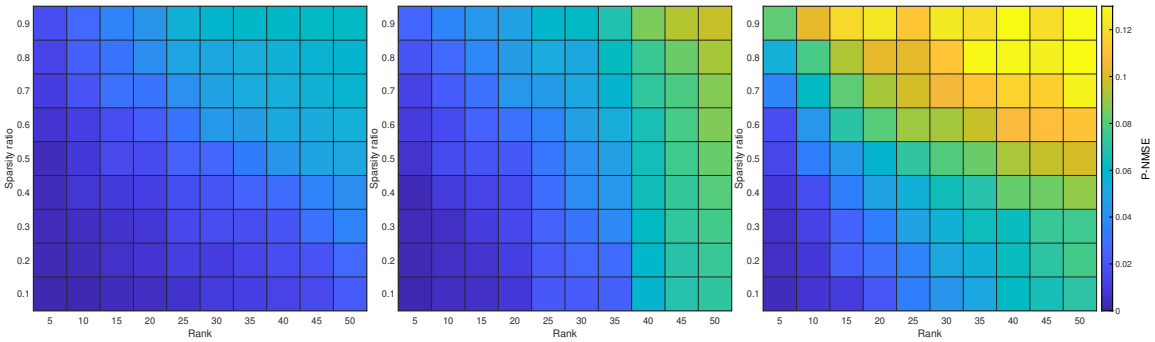
(a) BiG-VAMP-MC. (b) BiG-AMP-MC. (c) DA-niAPG.

Figure 5.2: P-NMSE of the MC problem on a randomly generated discrete matrix, $\boldsymbol{Y}$, with $\boldsymbol{U}$ and $\boldsymbol{V}$ being Gaussian, $N = 400$, $M = 300$ and $\delta = 30\%$ as a function of the sparsity ratio and rank values: (a) BiG-VAMP-MC, (b) BiG-AMP-MC, and (c) DA-niAPG.



(a) BiG-VAMP-MC. (b) BiG-AMP-MC. (c) DA-niAPG.

Figure 5.3: P-NMSE of the MC problem on a randomly generated discrete matrix, $\boldsymbol{Y}$, with $\boldsymbol{U}$ being binary, $\boldsymbol{V}$ being Gaussian, $N = 400$, $M = 300$ and $\delta = 30\%$ as a function of the sparsity ratio and rank values: (a) BiG-VAMP-MC, (b) BiG-AMP-MC, and (c) DA-niAPG.



(a) BiG-VAMP-MC. (b) BiG-AMP-MC. (c) DA-niAPG.

Figure 5.4: P-NMSE of the MC problem on a randomly generated discrete matrix, $\boldsymbol{Y}$, with $\boldsymbol{U}$ being binary, $\boldsymbol{V}$ being Bernoulli-Gaussian, with with a sparsity level of 95%, $N = 1000$, $M = 300$ and $\delta = 30\%$ as a function of the sparsity ratio and rank values: (a) BiG-VAMP-MC, (b) BiG-AMP-MC, and (c) DA-niAPG.

sets:

$$\boldsymbol{e} = \{e_1, e_2, e_3, e_4, e_5\},$$
$$\boldsymbol{r} = [-1.5\,\sigma_Z, -0.5\,\sigma_Z, 0.5\,\sigma_Z, 1.5\,\sigma_Z]^\top, \tag{5.1}$$

where, σ_Z is the standard deviation of the matrix $\boldsymbol{Z}$. In this synthetic experiment, we set the sparsity ratio $\rho = 60\%$ and the observed ratio $\delta = 30\%$, by illustrating the representation of the P-NMSE values as a function of iteration, when $\boldsymbol{U}$ is binary and $\boldsymbol{V}$ is Gaussian, and both $\boldsymbol{U}$ and $\boldsymbol{V}$ have Gaussian priors, in Fig. 5.1a and Fig. 5.1b, respectively. Both Fig. 5.1a and Fig. 5.1b show that BiG-VAMP-MC exhibits lower P-NMSE values in much fewer iterations as benchmarked to both APG and BiG-AMP-based approaches, it is also seen in Fig. 5.1a, all of the algorithms perform much better when the prior on $\boldsymbol{U}$ is binary rather than Gaussian, and all the compared methods were able to reach a better P-NMSE value in fewer iterations than the previous scenario. Here, both BiG-VAMP-MC and BiG-AMP-MC manage to successfully reconstruct all the non-observed 70% of the entries of $\boldsymbol{Y}$ without making any error in less than 10 and 30 iterations, respectively, where the other non-AMP-based methods were not able to perfectly reconstruct $\boldsymbol{Y}$ without making errors, this brings up one of the limitations of APG-based solutions, where they cannot handle discrete priors on $\boldsymbol{U}$ and/or $\boldsymbol{V}$ and only treat the observation data matrix as discrete, but in some cases, the discreteness can be extended to the factorized matrices for improved results. Following that, Fig. 5.2, Fig. 5.3 and Fig. 5.4 show the P-NMSE performance of BiG-VAMP-MC, BiG-AMP-MC, and DA-niAPG, over a two-dimensional grid of different sparsity ratio values and ranks, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are Gaussian, $\boldsymbol{U}$ is binary and $\boldsymbol{V}$ is Gaussian and $\boldsymbol{U}$ is binary and $\boldsymbol{V}$ is Bernoulli-Gaussian with a sparsity level of 95%, respectively. On the grid, it can be observed that more valid entries (i.e., smaller $\frac{|O|}{NM}$) improve the P-NMSE performance which shows a relatively smooth phase transition curve separating the success region in blue and the failure region orange. In addition, BiG-VAMP-MC performs better than all other state-of-the-art solutions, providing superior P-NMSE values in all three scenarios. The results are almost equivalent for both AMP-based methods in the case of continuous Gaussian priors on matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ in Fig. 5.2. However, BiG-VAMP-MC showed significant superior performance in both Fig. 5.3 and Fig. 5.4, where the prior on matrix $\boldsymbol{U}$ is binary. As demonstrated in our previous study [1], we found that BiG-AMP fails to converge when using discrete priors. Next, Fig. 5.5 displays the computational and theoretical MSE on the matrix $\widetilde{\boldsymbol{Y}}$ provided by BiG-VAMP and its SE, respectively. The experiment was conducted using a discrete matrix, $\boldsymbol{Y}$, with Gaussian $\boldsymbol{U}$ and $\boldsymbol{V}$, where $N = 1000$, $M = 500$, and $R = 5$. This

matrix was generated from the experiment settings used in (5.1). The error was tracked over time. There, it is observed that the empirical MSE coincides with the theoretical performance predicted by SE analysis after convergence. The delay in time is mostly due to the damping factor used to enhance the convergence behavior of BiG-VAMP. Finally, we computationally examine the effect of the rank R on the final quantization step size determined by the EM-threshold learning block in the case of 2 bits/symbol uniform quantization, using the synthetic observed matrix, $\boldsymbol{Y}$, with $N = 200$, $M = 200$, generated from Gaussian $\boldsymbol{U}$ and Gaussian $\boldsymbol{V}$ from the following sets:

$$\begin{aligned} \boldsymbol{e} &= \{e_1, e_2, e_3, e_4\}, \\ \boldsymbol{r} &= [-\Delta, 0, \Delta]^\top. \end{aligned} \tag{5.2}$$

This experiment can also give us insights into how the rank R impacts the correlation between the optimal learned quantization levels and the standard deviation. Therefore, by setting the quantization step size $\Delta = \sigma_Z$, the sparsity ratio $\rho = 40\%$ and the initial thresholds to $(r_1^0 = 1, r_2^0 = 2, r_3^0 = 3)$, it can be observed in Table 5.1 that the reconstruction error of the quantized data, along with the standard deviation of matrix $\boldsymbol{Z}$, increased with the increase in rank. Specifically, for lower ranks, the EM-threshold learning block returned an optimal uniform quantization step size within the same range that was initially used to quantize the data. However, as the rank increased, the value of the final quantization step, spreading the values of matrix $\boldsymbol{Z}$ into four uniform regions slightly changed compared to the initial one. This change occurred because the error of reconstructing the quantized matrix optimally became higher due to more unknown vectors to be estimated of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$.

5.3.2 Experiments on real-world data

For real-world applications, we use the observed discrete matrix $\boldsymbol{Y}$ from a very popular RSs dataset, known as the *MovieLens* dataset [38]. Precisely, its 100K version, which consists of a 943 (users) $\times$ 1682 (movies) rating matrix with a sparsity ratio $\rho = 93.7\%$, where at least 20 movies have been rated by each user, from 1-star to 5-star anonymous ratings. We display in Fig. 5.6a and Fig. 5.6b, the P-NMSE values for all considered discreteness-aware MF/MC baseline algorithms, on the MovieLens-100K when both $\boldsymbol{U}$ and $\boldsymbol{V}$ have Gaussian priors [3] for the low-

[3]The case when the matrix $\boldsymbol{U}$ has a binary prior and the matrix $\boldsymbol{V}$ is Gaussian, showed that the results were almost identical to the case where both matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are Gaussian. It is important to note that this specific outcome was observed only in the chosen dataset and settings, and may not hold in other scenarios.

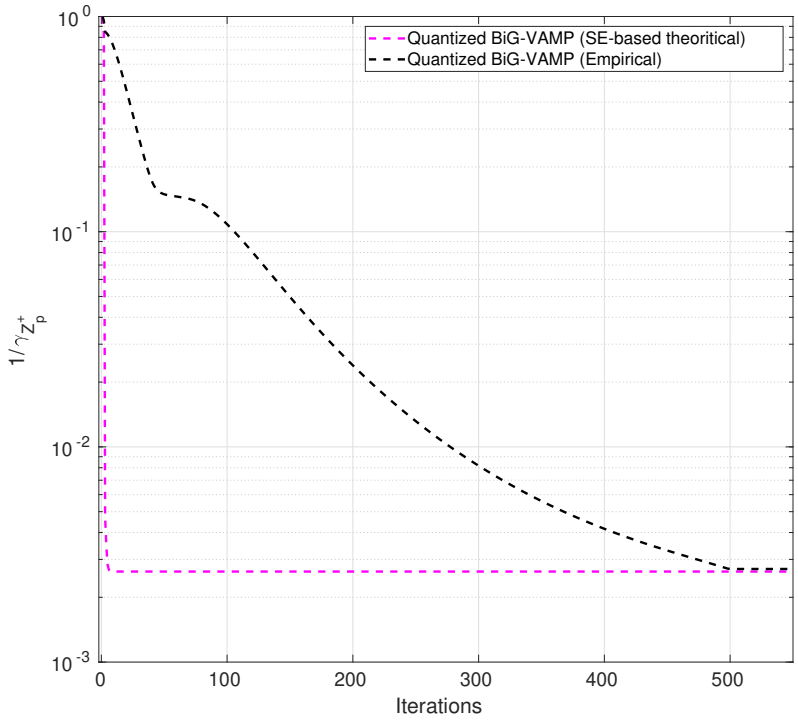


Figure 5.5: MSE on the output quantized denoiser for $\boldsymbol{Z}$ of BiG-VAMP-MC and its SE for the discrete-aware MC problem on a discrete matrix, $\boldsymbol{Y}$, with $\boldsymbol{U}$ and $\boldsymbol{V}$ being Gaussian, $N = 1000$, $M = 500$ and $R = 5$, generated according to the sets used in the experiment (5.1).

rank setting. For the learned BiG-VAMP-MC, we initialized the threshold values to $(r_1^0 = 1.5, r_2^0 = 2.5, r_3^0 = 3.5, r_4^0 = 4.5)$. After convergence, the optimal threshold values obtained were $(\widehat{r}_1^{T_{\max}} = -15.67, \widehat{r}_2^{T_{\max}} = -13.47, \widehat{r}_3^{T_{\max}} = 0.5, \widehat{r}_4^{T_{\max}} = 17.41)$, and $(\widehat{r}_1^{T_{\max}} = -14.32, \widehat{r}_2^{T_{\max}} = -12.56, \widehat{r}_3^{T_{\max}} = 2.29, \widehat{r}_4^{T_{\max}} = 19.63)$ for the simulations, in Fig. 5.6a and Fig. 5.6b, respectively. So as expected, with the exception of the non-quantized version of BiG-VAMP, all three of the other quantized algorithms were able to achieve a P-NMSE of less than 0.1 for both observed ratios with a significantly closer performance to each other. These results are notably superior to those of the majority of the latest existing RSs as of today. As discussed in 4.2.4, the suggested BiG-VAMP-based solution performs best when the threshold values are learned, highlighting the innovative additions of this research to the BiG-VAMP in terms of quantization and threshold learning. albeit these added novelties to the regular BiG-VAMP can reduce performance deterioration in both scenarios when 30% and 50% of the number of entries of the matrix can be viewed. For the DA-niAPG method, their rank regularizer returned a value of $R = 1$. This means that their performance relies on classifying all movies in

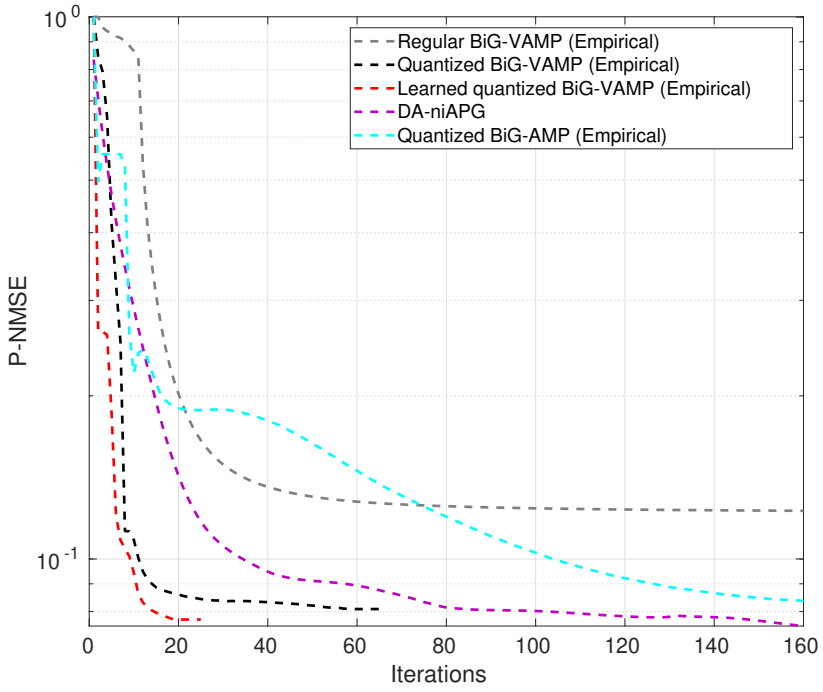
Table 5.1: True and final estimated threshold values, NMSE and the initial-final fraction values on a synthetically generated discrete matrix $\boldsymbol{Y}$, with $N = 200$, $M = 200$, $\boldsymbol{U}$ and $\boldsymbol{V}$ being Gaussian, over different rank values .

Rank	r_1	r_2	r_3	$\widehat{r}_1^{T_{\max}}$	$\widehat{r}_2^{T_{\max}}$	$\widehat{r}_3^{T_{\max}}$	NMSE
2	-1.3922	0	1.3922	-1.3086	-0.0765	1.3475	0.0036
5	-2.1709	0	2.1709	-1.5890	-0.1592	1.5895	0.0039
8	-2.7803	0	2.7803	-1.9849	0.2742	2.0023	0.0058
12	-3.4122	0	3.4122	-2.7933	-0.3956	2.7853	0.0065
15	-3.7991	0	3.7991	–4.1036	0.0286	4.1211	0.0070
18	-4.2146	0	4.2146	-5.1956	-0.0035	5.2188	0.0073
21	-4.4277	0	4.4277	–5.9894	-0.0047	5.8397	0.0084
24	-4.4783	0	4.4783	-7.9715	-0.0658	7.9123	0.0130
27	-4.8111	0	4.8111	-8.9065	0.9440	8.9205	0.0196
30	-5.4506	0	5.4506	-9.1263	0.1769	9.1527	0.0213

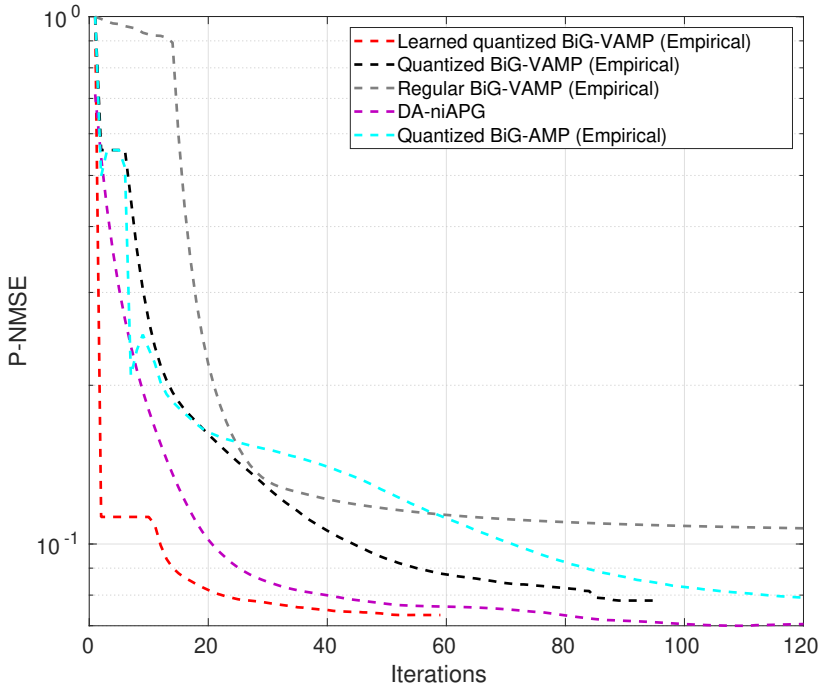
the MovieLens dataset into a single dominant category. In contrast, our BiG-VAMP-MC and BiG-AMP methods employ a rank of $R = 2$, which permits the classification of movies into two distinct dominant categories. It is worth mentioning that all three methods can perform well for higher ranks, but these particular rank values yielded the best results. It is also important to note that the APG-based framework solutions require extensive manual parameter adaptation that changes depending on the run setting and/or data set used. This is a significant limitation of APG-based solutions. Conversely, our method is an auto-learning one where the data set is the only input needed for the reconstruction and no extra parameter tuning is necessary, which is a significant advantage over APG-based solutions.

Conclusion

In this chapter, we presented several simulation experiments using both synthetic data and a movie recommendation system application. Our results demonstrate that BiG-VAMP-MC achieves the highest reconstruction accuracy compared to other existing discrete state-of-the-art algorithms such as BiG-AMP-MC and DA-niAPG. Additionally, we performed a SE analysis of BiG-VAMP-MC and demonstrated its consistency in the synthetic MC application.



(a) $\delta = 30\%$.



(b) $\delta = 50\%$.

Figure 5.6: P-NMSE of BiG-VAMP, BiG-AMP, and DA-niAPG for the discrete-aware MC problem on MovieLens-100K.

Conclusion

In this work, we presented a new quantized MC algorithm through bilinear factorization based on the BiG-VAMP framework, to address the discrete MF/MC problems. BiG-VAMP-MC uses output quantization and selection transformations, $\phi_q(\cdot)$ and $\phi_s(\cdot)$, respectively, to model discrete entries and handle missing observations in $\boldsymbol{Y}$. Furthermore, BiG-VAMP-MC takes advantage of the discrete-valued i.i.d. priors on $\boldsymbol{U}$ and $\boldsymbol{V}$, which are systematically handled by the BiG-VAMP framework.

In contrast to existing discreteness-aware methodologies, which only handle the observation data matrix as discrete, our new approach allows for the discretization of the factorized matrices for even better results. BiG-VAMP-MC is an AMP-like algorithm inspired by the BiG-VAMP framework, which can handle both discrete-valued entries in the observation matrix $\boldsymbol{Y}$ and the latent factors $\boldsymbol{U}$ and $\boldsymbol{V}$ simultaneously. Specifically, our numerical results using synthetic data and a movie RSs application exhibited that BiG-VAMP-MC has the best reconstruction accuracy as compared to existing discrete state-of-the-art algorithms such as BiG-AMP-MC and DA-niAPG. The algorithm further enhances performance in the RSs application by employing adaptive quantization thresholds of $\phi_q(\cdot)$, learned via the EM algorithm. Furthermore, we conducted a SE analysis of BiG-VAMP-MC and showed its consistency in the synthetic MC application.

References

[1] Akrout, Mohamed and Housseini, Anis and Bellili, Faouzi and Mezghani, Amine "Bilinear generalized vector approximate message passing", in 2022 56th Asilomar Conference on Signals, Systems, and Computers.

[2] F. Kschischang, J.Frey and H. Loeliger "Factor graphs and the sum-product algorithm", July 27 1998.

[3] Stanford CS228 online course, S. Ermon "Probabilistic graphical models".

[4] A. Montanari, Y. Eldar, and G. Kutyniok, "Graphical models concepts in compressed sensing", Compressed Sensing: Theory and Applications, pp. 394–438, 2012.

[5] A. Mezghani and J. A. Nossek, "Belief propagation based mimo detection operating on quantized channel output", in 2010 IEEE International Symposium on Information Theory. IEEE, 2010, pp. 2113–2117.

[6] M. Tulino, S. Verdúet al., "Random matrix theory and wireless communications", Foundations and Trends® in Communications and Information Theory, vol. 1, no. 1, pp. 1–182, 2004.

[7] R. Matsushita and T. Tanaka, "Low-rank matrix reconstruction andclustering via approximate message passing", in Advances in Neural Information Processing Systems, 2013, pp. 917–925.

[8] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing", Proceedings of the National Academy of Sciences, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[9] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing", IEEE Transactions on Information Theory, 2019.

[10] R. Gribonval, R. Jenatton, and F. Bach, "Sparse and spurious: dictionary learning with noise and outliers", IEEE Transactions on InformationTheory, vol. 61, no. 11, pp. 6298–6319, 2015.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems", Computer, no. 8, pp. 30–37, 2009.

[12] Y. Jin Lim, Y. Whye Teh, "Variational Bayesian Approach to Movie Rating Prediction", KDDCup.07 August 12, 2007.

[13] . Schniter, "BiGAMP", [Online]. Available: https://sourceforge.net/projects/gampmatlab/.

[14] S. Sarkar, "Bilinear Adaptive Vector Approximate Message Passing", Jul. 2019. [Online]. Available: https://github.com/sbrsarkar/BAdVAMP.

[15] F. Krzakala, "Low rank Matrix Factorization with AMP", Aug. 2015.[Online]. Available: https://github.com/krzakala/LowRAMP.

[16] Kueng, Richard and Tropp, Joel A, "Binary component decomposition Part I: the positive-semidefinite case", 2021. SIAM Journal on Mathematics of Data Science.

[17] Agarwal, Deepak and Chen, Bee-Chung, "Regression-Based Latent Factor Models", 2009. Proceedings of Machine Learning Research.

[18] Kim, Yong-Deok and Choi, Seungjin, "Scalable Variational Bayesian Matrix Factorization with Side Information", 2014. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[19] Rai, Piyush, "Non-Negative Inductive Matrix Completion for Discrete Dyadic Data", 2017. Proceedings of the AAAI Conference on Artificial Intelligence.

[20] Huo, Zhouyuan and Liu, Ji and Huang, Heng, "Optimal Discrete Matrix Completion", 2016. Proceedings of the AAAI Conference on Artificial Intelligence.

[21] Iimori, Hiroki and de Abreu, Giuseppe Thadeu Freitas and Taghizadeh, Omid and Ishibashi, Koji, "Discrete-Aware Matrix Completion via Proximal Gradient", 2020. 54th Asilomar Conference on Signals, Systems, and Computers.

[22] Dorffer, Clément and Puigt, Matthieu and Delmaire, Gilles and Roussel, Gilles, "Blind mobile sensor calibration using an informed nonnegative matrix factorization with a relaxed rendezvous mode", 2016. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[23] Zhang, Shuai and Yao, Lina and Sun, Aixin and Tay, Yi, "Deep learning based recommender system: A survey and new perspectives", 2019. ACM Computing Surveys (CSUR).

[24] Lee, Daniel D and Seung, H Sebastian, "Learning the parts of objects by non-negative matrix factorization", 1999. Nature.

[25] Lee, Joonseok and Sun, Mingxuan and Lebanon, Guy, "A comparative study of collaborative filtering algorithms", 2012. arXiv preprint arXiv:1205.3193.

[26] Sarwar, Badrul and Karypis, George and Konstan, Joseph and Riedl, John, "Item-based collaborative filtering recommendation algorithms", 2001. Proceedings of the 10th international conference on World Wide Web.

[27] Lee, Joonseok and Kim, Seungyeon and Lebanon, Guy and Singer, Yoram and Bengio, Samy, "LLORMA: Local low-rank matrix approximation", 2016. The Journal of Machine Learning Research.

[28] Goldberg, Ken and Roeder, Theresa and Gupta, Dhruv and Perkins, Chris, "Eigentaste: A constant time collaborative filtering algorithm", 2001. information retrieval.

[29] Ronen, Royi and Koenigstein, Noam and Ziklik, Elad and Sitruk, Mikael and Yaari, Ronen and Haiby-Weiss, Neta, "Sage: recommender engine as a cloud service", 2013. Proceedings of the 7th ACM conference on Recommender systems.

[30] Roweis, Sam T and Saul, Lawrence K, "Nonlinear dimensionality reduction by locally linear embedding", 2000. American Association for the Advancement of Science.

[31] Belkin, Mikhail and Niyogi, Partha, "Laplacian eigenmaps and spectral techniques for embedding and clustering", 2002. Advances in neural information processing systems.

[32] Sharifi, Zeinab and Rezghi, Mansoor and Nasiri, Mahdi, "A new algorithm for solving data sparsity problem based-on Non negative matrix factorization in recommender systems", 2014. 4th International Conference on Computer and Knowledge Engineering (ICCKE).

[33] Zhang, Zhongyuan and Li, Tao and Ding, Chris and Zhang, Xiangsun, "Binary Matrix Factorization with Applications", 2007. Seventh IEEE International Conference on Data Mining (ICDM 2007).

[34] Zhang, Hanwang and Shen, Fumin and Liu, Wei and He, Xiangnan and Luan, Huanbo and Chua, Tat-Seng, "Discrete Collaborative Filtering", 2016. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval).

[35] Liu, Chenghao and Wang, Xin and Lu, Tao and Zhu, Wenwu and Sun, Jianling and Hoi, Steven, "Discrete Social Recommendation", 2019. Proceedings of the AAAI Conference on Artificial Intelligence.

[36] LMatsushita, Ryosuke and Tanaka, Toshiyuki, "Low-rank matrix reconstruction and clustering via approximate message passing", 2013. Advances in Neural Information Processing Systems.

[37] Parker, Jason T and Schniter, Philip and Cevher, Volkan, "Bilinear generalized approximate message passing—Part I: Derivation", 2014. IEEE Transactions on Signal Processing.

[38] Harper, F. Maxwell and Konstan, Joseph A, Volkan, "The MovieLens Datasets: History and Context", 2015. ACM Trans. Interact. Intell. Syst.

[39] Toh, Kim-Chuan and Yun, Sangwoon, "An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Least Squares Problems", 2010. Pacific Journal of Optimization 10.1145/2827872.

[40] Moon, T.K., "The expectation-maximization algorithm", 1996. IEEE Signal Processing Magazine 10.1109/79.543975.

[41] Gribonval, Rémi and Jenatton, Rodolphe and Bach, Francis, "Sparse and spurious: dictionary learning with noise and outliers", 2015. IEEE Transactions on Information Theory.

[42] Montanari, Andrea and Eldar, YC and Kutyniok, G, "Graphical models concepts in compressed sensing", 2012. Compressed Sensing: Theory and Applications.

[43] Zou, Hui and Hastie, Trevor and Tibshirani, Robert, "Sparse principal component analysis", 2006. Journal of computational and graphical statistics.

[44] Koren, Yehuda and Bell, Robert and Volinsky, Chris, "Matrix factorization techniques for recommender systems", 2009. Computer, IEEE.

[45] Wang, Lu and Zhang, Zhengwu and Dunson, David, "Symmetric Bilinear Regression for Signal Subgraph Estimation", 2019. IEEE Transactions on Signal Processing.

[46] Mezard, Marc and Montanari, Andrea, "Information, physics, and computation", 2009. Oxford University Press.

[47] Pech, Ratha and Hao, Dong and Pan, Liming and Cheng, Hong and Zhou, Tao, "Link prediction via matrix completion", 2017. EPL (Europhysics Letters), IOP Publishing.

[48] Li, Bo and Petropulu, Athina P. and Trappe, Wade. "Optimum Co-Design for Spectrum Sharing between Matrix Completion Based MIMO Radars and a MIMO Communication System", 2016. IEEE Transactions on Signal Processing.

[49] Sun, Shunqiao and Bajwa, Waheed U. and Petropulu, Athina P. "MIMO-MC radar: A MIMO radar approach based on matrix completion", 2015. IEEE Transactions on Aerospace and Electronic Systems.

[50] Kang, Zhao and Peng, Chong and Cheng, Qiang, "Top-N Recommender System via Matrix Completion", 2016. https://ojs.aaai.org/index.php/AAAI/article/view/9967.

[51] Ramlatchan, Andy and Yang, Mengyun and Liu, Quan and Li, Min and Wang, Jianxin and Li, Yaohang, "A survey of matrix completion methods for recommendation systems", 2018. Big Data Mining and Analytics.

[52] Xie, Kun and Ning, Xueping and Wang, Xin and Xie, Dongliang and Cao, Jiannong and Xie, Gaogang and Wen, Jigang, "Recover Corrupted Data in Sensor Networks: A Matrix Completion Solution", 2017. IEEE Transactions on Mobile Computing.

[53] Liu, Zhang and Vandenberghe, Lieven, "Interior-Point Method for Nuclear Norm Approximation with Application to System Identification", 2010. SIAM Journal on Matrix Analysis and Applications.

[54] Bennett, James and Lanning, Stan and others, "The netflix prize", 2007. Proceedings of KDD cup and workshop.

[55] Natarajan, Nagarajan and Dhillon, Inderjit S, "Inductive matrix completion for predicting gene–disease associations", 2014. Bioinformatics.

[56] Davenport, Mark A. and Plan, Yaniv and van den Berg, Ewout and Wootters, Mary, "1-Bit matrix completion", 2014. Information and Inference: A Journal of the IMA.

[57] T. Tony Cai and Wen-Xin Zhou, "A Max-Norm Constrained Minimization Approach to 1-Bit Matrix Completion", 2013. stat.ML.

[58] Ni, Renkun and Gu, Quanquan, "Optimal Statistical and Computational Rates for One Bit Matrix Completion", 2013. Proceedings of the 19th International Conference on Artificial Intelligence and Statistics.

[59] Sonia A. Bhaskar, "Probabilistic Low-Rank Matrix Completion from Quantized Measurements", 2016. Journal of Machine Learning Research.

[60] Ashkan Esmaeili and Kayhan Behdin and Sina Al-E-Mohammad and Farokh Marvasti, "Recovering Quantized Data with Missing Information Using Bilinear Factorization and Augmented Lagrangian Method", 2018. stat.ML.

[61] Soleymani, Mahdi and Liu, Qiang and Mahdavifar, Hessam and Balzano, Laura, "Matrix Completion over Finite Fields: Bounds and Belief Propagation Algorithms", 2023 IEEE International Symposium on Information Theory (ISIT).

[62] Schniter, Philip and Rangan, Sundeep and Fletcher, Alyson, "Denoising based vector approximate message passing", 2016 arXiv preprint arXiv:1611.01376.

[63] Pincus, Martin, "Errata: A Closed Form Solution of Certain Programming Problems", 1969 Operations Research.