Scalable Bayesian Inversion with Poisson Data

Chen Zhang

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

April 14, 2020

I, Chen Zhang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Poisson data arise in many important inverse problems, e.g., medical imaging. The stochastic nature of noisy observation processes and imprecise prior information implies that there exists an ensemble of solutions consistent with the given Poisson data to various extents. Existing approaches, e.g., maximum likelihood and penalised maximum likelihood, incorporate the statistical information for point estimates, but fail to provide the important uncertainty information of various possible solutions. While full Bayesian approaches can solve this problem, the posterior distributions are often intractable due to their complicated form and the curse of dimensionality. In this thesis, we investigate approximate Bayesian inference techniques, i.e., variational inference (VI), expectation propagation (EP) and Bayesian deep learning (BDL), for scalable posterior exploration.

The scalability relies on leveraging 1) mathematical structures emerging in the problems, i.e., the low rank structure of forward operators and the rank 1 projection form of factors in the posterior distribution, and 2) efficient feed forward processes of neural networks and further reduced training time by flexibility of dimensions with incorporating forward and adjoint operators.

Apart from the scalability, we also address theoretical analysis, algorithmic design and practical implementation. For VI, we derive explicit functional form and analyse the convergence of algorithms, which are long-standing problems in the literature. For EP, we discuss how to incorporate nonnegative constraints and how to design stable moment evaluation schemes, which are vital and nontrivial practical concerns. For BDL, specifically conditional variational auto-encoders (CVAEs), we investigate how to apply them for uncertainty quantification of inverse problems and develop flexible and novel frameworks for general Bayesian Inversion.

Finally, we justify these contributions with numerical experiments and show the competitiveness of our proposed methods by comparing with state-of-the-art benchmarks.

Impact Statement

In this thesis, we studied Bayesian inference for inverse problems with Poisson data. By investigating how to design scalable algorithms with specific characteristics emerging in real world applications, we enabled uncertainty quantification for related large scale real world problems. Apart from scalability, we also pushed forward the frontier by deepening theoretical understandings, discussing how to incorporate practical constraints and building general and flexible frameworks.

Inside academia, we make the uncertainty information available in an efficient way so that downstream research in the application communities, e.g., the medical imaging community, on how to leverage the uncertainty information is doable. Researchers in the machine learning community with interest in theoretical analysis of related probabilistic frameworks could benefit from the ideas and results we showed. Besides, the methodologies and frameworks developed in this project could also be investigated for other research in inverse problems, e.g., Photo-Acoustic imaging, MR imaging, etc.

Outside academia, the research results in this thesis have potential to be applied to related applications, i.e., PET imaging clinics. Our frameworks could provide not only point estimates of the patients' inner body information but also the associated uncertainty for a certain measurement. Such uncertainty information is not available with current methods in production and vital for highly noisy observations, e.g., low-dose imaging. With our provided uncertainty information, clinicians may be aware of more objective and comprehensive information to conduct diagnosis.

Acknowledgements

I would like to extend my heartfelt gratitude to my supervisors, Prof. Bangti Jin and Prof. Simon Arridge, whose inspiring and fruitful supervisions have guided my way of being a researcher. Without their profound knowledge and support, I would not have been capable to conquer the difficulties and sail towards the destination of this project.

Bangti has never stopped surprising me with his ability to see the essence through complicated and perplexing surfaces. His very intelligence, intuition and insight make every discussion between us enjoyable and effective. Working with him has opened a new dimension for me to identify and investigate research problems.

Simon has always been impressing me with his attitude towards detail and excellence. His abundant knowledge of established theories and curiosity towards emerging subjects have shown me a way how to successively success in a field.

I would give my sincerest thanks to my family for their persistent love and support. They have cultivated my interest towards exploring unknowns since my childhood, which points me the direction I am now pursuing. It is their enlightenment and encouragement that starts my dream of being a researcher.

I would also like to thank the UCL Computer Science Department for funding my research, which together with above mentioned and my own efforts realise the dream.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Poisson data widely arise in phenomena involving number counting, which include a large spectrum of real world applications, including medical imaging, neural science and quantitative finance to name a few. From the highest level of view, in these scenarios, 1) we can observe the counting variable y, modelled by Poisson distributions and resulted from the variable of interest x, and 2) we aim to reconstruct the unobservable x from the counting observation y. Specifically, each entry y_i in the observable variable $y = [y_i]_i$ (a compact notation for a vector with index i and of an implied length) follows a Poisson distribution

$$p(y_i|x) = \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!},$$

where $\lambda_i = g_i(x)$ specifies the mean and variance of the Poisson distribution. The task of inverse problems with Poisson data is to reconstruct the variable x which leads to the observation y through above Poisson distributions.

While the Poisson distribution acts as the source of noise in the observation process, the functions $\{g_i(x)\}_i$ encode the domain knowledge explaining how the unobservable variable x is transformed to some noise free $\lambda = [\lambda_i]_i$ in the observation space. In many applications, each $g_i(x)$ is a composition function with two components like one layer in neural networks. It consists of a linear transformation of x followed by an activation function, often referred to as inverse link function in the statistics literature. In this section, we will review the probabilistic models of X-ray computational tomography (X-ray CT) and positron emission tomography (PET) where we have complicated inverse link functions with exponential form and simple ones with linear form in $\{g_i(x)\}_i$, respectively. X-ray CT and PET are respectively representative examples of transmission tomography and emission tomography. By reviewing these two important medical imaging modalities and current approaches to solve associated inverse problems, we not only stress the importance and wide appearance of Poisson data, but also motivate the research in this thesis.

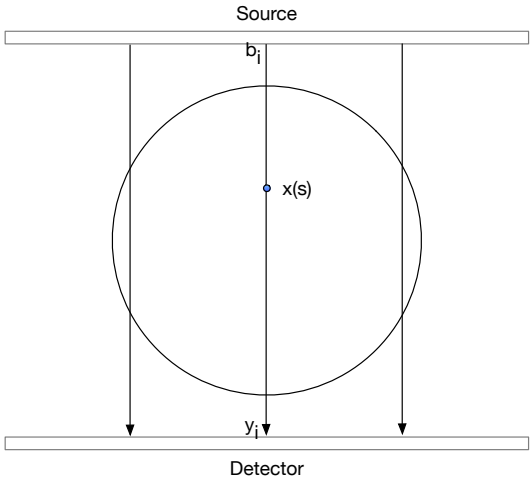


Figure 1.1: The illustration of transmission tomography

In transmission tomography, e.g., X-ray CT, the source of the radiation is outside of the patient and the intensity of the radiation will decay as it goes through the patient. The decay of the intensity is determined by the attenuation coefficients, which form the variable of interest x, of the patient. By observing the decayed intensity of the radiation, which is y, one can reconstruct the attenuation coefficients with proper forward models.

The forward model describing radiation attenuation is based on the Beer-Lambert law. To explain, consider a bounded two-dimensional area parameterised by $s = (s_1, s_2)$, which defines the domain of a patient's inner body. The attenuation coefficients can be represented by a function of s, denoted by $x = x(s)$. Consider a beam of light I_i whose intensity at the source is b_i and which goes through the line $L_i(s)$. The Beer-Lambert law asserts, after attenuation in the domain, the intensity would be $y_i = b_i \exp(-\int_{L_i} x(s)\mathrm{d}s)$. In a certain angle, there can be many beams of light and each beam of light gives one projection y_i. Due to the rotation of the imaging equipment, there can be many angles. If we stack all y_i's from different angles and beams, we have the observation y.

However, this deterministic physical model is an idealised model overlooking many realistic factors, e.g., the stochastic properties of the detectors, background events and so on. The probabilistic forward model of transmission tomography regards the observation $y = [y_i]_i$ as a sequence of Poisson random variables with means $\{\mathbf{E}[y_i] = b_i \exp(-\int_{L_i} x(s)\mathrm{d}s) + r_i\}_i$ where $r = [r_i]_i$ is the intensity of background events. Note that although the background events $\{r_i\}_i$ can also be interpreted as additive noise, the real random noise modelled by Poisson distributions (not the background events) happened at detectors is neither additive nor multiplicative. Since the problem of reconstructing x cannot be directly solved numerically in the continuous setting, the bounded area is usually discretised into grids. To enable the matrix-vector representation of Radon transform, we would flatten discretised x into a column vector. Accordingly, the line integrals (Radon transform) can be represented by a matrix A. Further, the mean vector of the forward model is $b \odot e^{Ax} + r$, where $\odot$ is the

element-wise Hadamard product.

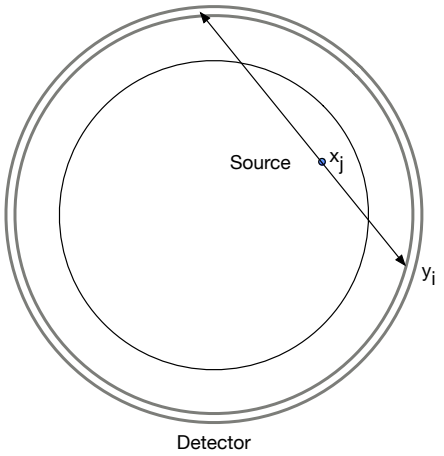


Figure 1.2: The illustration of emission tomography

In emission tomography, e.g., PET and SPECT (single photon emission computed tomography), the source of radiation is inside the patient. Here we would use the mechanism of PET as an example for probabilistic model building. Radiopharmaceuticals, materials containing radioactive isotopes, are introduced into a patient's body which would emit positrons during radioactive decay. When a positron is emitted at an inner point of the body, it will annihilate with a nearby electron and creates a pair of gamma photons going off two opposite directions. Then a pair of detectors at the ends of these two opposite directions will observe these two photons at the same time if we neglect the attenuation and scattering. Denote the intensity of the γ-ray detected by the i-th detector pair by y_i. The variable of interest x is a function defined on the bounded area, which is parameterised by s, recording the intensity of emissions at each point in the domain.

Since a pair of photons can go off any two opposite directions, deterministic physical models are not sufficient to explain the phenomenon. Let $x = [x_j]_j$ be the discretised intensity function vector and A_{ij} be the probability of a photon pair from the position x_j detected by the i-th detector pair. Considering the background events with intensity $r = [r_i]_i$, we can model the expectation of the γ-ray intensity detected by the detector pairs by $\mathbf{E}[y] = Ax + r$. Since each row of A is a probability simplex, the sum of each row is one. In reality, the probabilities will be corrected by taking attenuation and scattering into consideration, which means the rows of A are not necessarily sum-one.

With the examples of X-ray CT and PET, we see how the Poisson distribution could be used to model the stochastic forward process in real world problems. Due to the presence of random noise in the forward process, a single unobservable ground truth could lead to a set of different possible observations. And different sets of possible observations resulted by different ground truths may have non-empty intersections. This means that for a single observation, there are various ground truths being able to explain the observation with the probabilistic forward model. Specifically for

medical imaging, there might be different conditions of the patient which could lead to the same observation received by a medical scanner. When a clinician is conducting diagnosis of a patient based on the information provided by the medical scanner, a single reconstructed image may lead to biased decision makings.

In the literature of medical imaging, most works, either on transmission tomography [43], emission tomography [33, 45, 44, 89] or more general inverse problems [3], are based on maximum likelihood estimate (MLE) or penalised maximum likelihood estimate (Penalised MLE). It is worth noting that the penalisation term acts as a regulariser and the Penalised MLE objective functional recovers Tikhonov regularisation in the classical inverse problems theory. Although these methods take the statistical information of forward models into consideration, they only focus on point estimates, which most probably lead to the Poisson observations. As a result, the important information of uncertainty quantification is not available from these methods. In contrast, in this thesis, we would explore full Bayesian methods which could capture the information of the whole posterior distributions. For more discussions on Poisson models in this avenue, we refer interested readers to recent surveys [111, 17, 64].

The Bayesian framework provides a systematic framework to facilitate uncertainty quantification for inverse problems [73, 128]. Besides treating y as a random variable and specifying a probabilistic forward model $p(y|x)$, the Bayesian framework would also treat x as a random variable and incorporate *a priori* information of it into the prior distribution $p(x)$. Then the Bayesian belief of possible unobservables are encoded in the posterior distribution $p(x|y)$, which, by Bayes' rule, is given by

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)\mathrm{d}x}.$$

The Bayesian belief of possible unobservables values the probabilities of possible ground truths due to the presence of noise and weighs them by the prior information. One can recover Penalised MLE by considering maximum *a posteriori* (MAP) estimate with a suitable prior distribution. For example, ℓ_2 penalised MLE is equivalent to MAP with a Gaussian prior. Similarly, one can recover MLE with uniform prior belief. Although the Bayesian framework could provide more comprehensive information, the posterior distribution $p(x|y)$ is often intractable. Even for the likelihood function and prior distribution with explicit forms, the evaluations of $p(y) = \int p(y|x)p(x)\mathrm{d}x$, a.k.a. the evidence, and other summarising statistics, e.g., mean and variance, suffer from the curse of dimensionality. Thus, sophisticated techniques, for exploring the posterior distributions up to a normaliser in a scalable manner, are needed.

In the literature of computational statistics and machine learning, popular Bayesian inference techniques are categorised into two folds, i.e., Monte Carlo sampling methods and approximate inference methods. Monte Carlo sampling methods are developed to generate samples from a distribution to approximate integrals and not restricted to posterior distributions. To name a few, we

have reject sampling, importance sampling, Metropolis-Hastings sampler, Gibbs sampling, Hamilton Monte Carlo sampling, etc. Since many sampling methods do not require the complete form of the target distribution, i.e., distributions up to a normalising constant are also applicable, they are very suitable for posterior exploration where we have the intractable evidence. Furthermore, they enjoy both practical and theoretical accuracy. Thus, sampling methods are most popular and well-developed computational methods for Bayesian inference in the statistics community. To sketch the ideas and results about sampling methods, we review a classical and representative MCMC algorithm, i.e., Metropolis-Hastings sampler.

The key ingredient of the Metropolis-Hastings algorithm is the proposal step and ratio based acceptance step. To construct a Markov chain $\{x_t\}$ for the target distribution $p(x)$, in the t-th iteration, we first generate a candidate x from a proposal distribution $q(x|x_{t-1})$, where x_{t-1} is the sample accepted in the last iteration. After the proposal step, we calculate the acceptance ratio $\rho = \min\{\frac{p(x)q(x_{t-1}|x)}{p(x_{t-1})q(x|x_{t-1})}, 1\}$. Then we generate a uniform random number α and accept the candidate x as x_t if $\rho \leq \alpha$. For symmetric proposal distributions, i.e., $q(x|x_{t-1}) = q(x_{t-1}|x)$, the acceptance ratio is simplified to be $\rho = \min\{\frac{p(x)}{p(x_{t-1})}, 1\}$. Note that the computation of ρ may suffer from numerical issues. Therefore, the evaluation is normally conducted in the logarithmic domain. Since any multiplicative constant in the target distribution $p(x)$ will be cancelled in ρ, we can apply the algorithm to joint distribution directly.

Among all the sampling methods, Markov chain Monte Carlo (MCMC) methods represent an important class of them. By definition, any methods constructing an ergodic Markov chain with stationary distribution the same as the target distribution is a MCMC method [119]. It is shown that Metropolis-Hasting is a MCMC method and thus has asymptotic convergent properties. However, the diagnosis of the chain is not straightforward. In practice, one can record traces of the chain or calculate the auto-correlations of the chain. If the traces have no obvious periodic shape or the auto-correlations are very small, one can regard the chain as achieving convergence.

Following the generic Metropolis-Hastings algorithm, many advances in the regime of Monte Carlo methods are developed to address related theoretical and practical concerns. Concurrently, the research has been extended to models related to the scope of this thesis. For example, Durmus *et al.* [38] leveraged the tools of convex analysis and discussed MCMC with Langevin dynamics for problems with non-smooth priors, which are common for imaging problems. Vargas *et al.* [132] further improved this idea by using a Runge-Kutta-Chebyshev stochastic approximation rather than the conventional Euler-Maruyama scheme. Besides, two very recent work [133, 145] investigated modern MCMC algorithms specifically for Poisson data. In contrast, another branch of Bayesian inference, i.e., approximate inference methods, is less explored. In this thesis, we investigate approximate inference methods, predominantly variational inference, expectation propagation and Bayesian deep learning, for inverse problems with Poisson data as alternative approaches to MCMC techniques. For readers interested in advances of MCMC, we refer to two recent review papers [56, 106] and

references therein. It is worth noting that apart from above two branches of full Bayesian methods, recent works [105, 24, 115] investigate how to leverage convex optimisation to obtain the MAP estimate together with a Highest Posteriori Density (HPD) interval. Although the posterior distributions are not fully recovered by this line of work, they could provide uncertainty information to a certain level and thus can be used for uncertainty quantification of log-concave posterior distributions, which covers several classical models in imaging problems.

To conclude the problem statement, we recall that the uncertainty of interest in this thesis is the uncertainty of the unobservable variable conditioned on the observation due to the randomness in the forward process and stochastic prior belief on the unobservable and we focus on the computational perspective of uncertainty quantification.

In the literature of uncertainty quantification and machine learning, three kinds of uncertainty are investigated, e.g., aleatoric uncertainty [77, 85], epistemic uncertainty [48, 77] and distributional uncertainty [95]. Aleatoric uncertainty, a.k.a., data uncertainty, is the uncertainty intrinsic in the data or in natural phenomenon, which can not be explained away with more data. A representative and naive example of aleatoric uncertainty is die rolling, i.e., for an even die, no matter how many times we roll it, the uncertainty of the next outcome will not change. Epistemic uncertainty, a.k.a., model uncertainty, is the uncertainty of the model to explain data, which is reducible and can be explained away with more data. Continue with the example of the die rolling, if we do not know the distribution of the die's outcome and we would like to come up with a model. A natural way to build a model is to use the empirical distribution from rolling test and the more times we roll the die, the more certain we would be about the empirical model. Distributional uncertainty is often referred to as unknown-unknown, which the uncertainty caused by the deviation of training and test data. It could be regarded as a self-diagnosis of a model's generalisation. When a sample not from the training dataset is input into the model, the alert could be raised by high distributional uncertainty indicating the model is not certain about the prediction since the input is out of the distribution of training data. From this perspective, the uncertainty studied in this thesis can be classified into aleatoric uncertainty or more specifically, conditional aleatoric uncertainty.

In addition to the computational perspective, many other perspectives and topics in the spectrum of uncertainty quantification for Bayesian inverse problems still remain challenging. Along the Bayesian pipeline, the first open problem is how to analyse model misspecification, which includes forward model misspecification and prior misspecification. Then it comes how to compute the uncertainty information encoded in the posterior distribution, which is the main focus of this thesis. Finally, how to fully leverage the uncertainty information uncovered with computational methods to aid real applications still calls for further investigation. Although we conduct this research project with the computational perspective, potential future works within the wider Bayesian context are discussed in Chapter 5.

1.2 Literature Review

In the last section, we reviewed two representative imaging modalities with Poisson data and current approaches to associated inverse problems. While current approaches leverage the statistical information in forward models, they only provide point estimates and neglect the important uncertainty information. In this section, we will start the investigation on scalable Bayesian inference techniques by reviewing another branch of posterior exploration schemes in the machine learning community, i.e., approximate inference methods.

The general idea of approximate inference is to find a tractable distribution to approximate the intractable posterior distribution. The approximation could be found by solving a direct variational problem between the approximation and the true posterior distribution, e.g., variational inference [26, 78, 103, 120], or iteratively updated by solving indirect variational problems between the approximation and some intermediary distributions, e.g., expectation propagation [99, 98]. More recently, deep neural networks (DNNs) are introduced to the approximation procedure and form a branch of Bayesian deep learning [48, 79, 80, 126]. We shall show the ideas of these approximate inference techniques and discuss potential problems applying them to inverse problems with Poisson data following this roadmap.

1.2.1 Variational Inference

1.2.1.1 General Framework

Variational inference (VI) methods formulate the approximation problem by some variational problem which selects the approximation by minimising some probabilistic metric, e.g. f-divergences [31, 5], Bregman divergences [22], etc. Despite the existence of various probabilistic metrics, VI with KL divergence [83] is most popular and well-developed due to the simple form and good interpretability. Recent works [88, 61] in the machine learning community also extend the discussion to α-divergence [6, 8, 114] which generalises KL divergence and belongs to both f-divergences and Bregman divergences. In this subsection, we will use KL divergence as an example to review the idea of variational inference.

The KL divergence $\mathrm{KL}(p||q)$ of two probabilistic densities $p(x)$ and $q(x)$ is defined by

$$\mathrm{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x. \tag{1.1}$$

By Jensen's inequality, we can see that $\mathrm{KL}(p||q) \geq 0$. By the property of integral, it vanishes if and only if $p(x) = q(x)$ almost everywhere. Moreover, we do not have $\mathrm{KL}(p||q) = \mathrm{KL}(q||p)$, for all $p(x)$ and $q(x)$, and thus KL divergence is not a mathematical metric.

Variational inference with KL divergence finds the approximation $q(x|y)$ to the true posterior

distribution $p(x|y)$ through the variational problem

$$q(x|y) = \arg\min_{\tilde{q}(x|y)\in\mathcal{Q}} \mathrm{KL}(\tilde{q}(x|y)||p(x|y)). \tag{1.2}$$

Due to the asymmetry of KL divergence, KL optimisation problems of two different directions will normally give different approximations [19]. Not rigorously, $\min_q \mathrm{KL}(p||q)$ tends to select q which puts more density on the area where p is large and thus corresponds to a mode-seeking manner [19, Section 10.1.2]. On the contrary, $\min_q \mathrm{KL}(q||p)$ tends to select q which has less density on the area where p is small and thus corresponds to a zero-avoiding manner. In this sense, variational inference with KL divergence would give an approximation concentrating in the neighbourhood of a mode of the posterior distribution. Since MAP is sometimes a better point estimate than mean in a posterior distribution, especially when the distribution is highly skewed, variational inference is especially suitable for such scenarios or single mode problems.

In general, there are two factors defining a variational problem, i.e., the variational metric and the variational family. Although we mainly focus on KL divergence, different variational families would lead to different algorithms. Different variational families include but not limit to mean-field Gaussian distributions [134], multivariate Gaussian distributions [26], Stein variational families [91], etc. The choice of variational families is often a trade off between efficiency and expressibility. For example, variational inference with mean field Gaussian family is more efficient to implement than that with multivariate Gaussian family but will overlook the covariance information. While efficiency is important in statistical computing, we cannot sacrifice the expressibility of the variational family. When the posterior distribution is very complicated, often satisfactory approximations do not belong to simple variational families. Thus, it is highly urgent to explore how to reduce the computational complexity for variational families with better expressibility, which is one of the contributions of our first work.

Substituting the posterior distribution $p(x|y)$ with the formula given by Bayes' theorem, we have

$$q(x|y) = \arg\min_{\tilde{q}(x|y)\in\mathcal{Q}} \int \tilde{q}(x|y)\log\frac{\tilde{q}(x|y)}{Z^{-1}p(y|x)p(x)}\mathrm{d}x. \tag{1.3}$$

Since the normaliser $Z(y) = \int p(y|x)p(x)\mathrm{d}x$ is unknown and does affect the solution to the problem, this variational problem is still intractable. Observe that

$$\log Z(y) = \int \tilde{q}(x|y)\log\frac{p(y|x)p(x)}{\tilde{q}(x|y)}\mathrm{d}x + \int \tilde{q}(x|y)\log\frac{\tilde{q}(x|y)}{p(x|y)}\mathrm{d}x, \tag{1.4}$$

we can turn to maximise the functional

$$F(\tilde{q}(x|y), p(y|x), p(x)) = \int \tilde{q}(x|y)\log\frac{p(y|x)p(x)}{\tilde{q}(x|y)}\mathrm{d}x. \tag{1.5}$$

Since $\mathrm{KL}(\tilde{q}(x|y)||p(x|y)) \geq 0$, the functional F acts as a lower bound of the logarithm of the evidence $Z(y)$. Hence, F is often referred to as the evidence lower bound (ELBO). To summarise, the variational problem is eventually reduced to

$$q(x|y) = \arg\max_{\tilde{q}(x|y)\in\mathcal{Q}} \int \tilde{q}(x|y) \log \frac{p(y|x)p(x)}{\tilde{q}(x|y)} \mathrm{d}x. \tag{1.6}$$

It is worth noting that the optimisation of ELBO admits a regularisation interpretation. Notice that F can be equivalently written as

$$F(\tilde{q}(x|y), p(y|x), p(x)) = \int \tilde{q}(x|y) \log p(y|x) \mathrm{d}x + \int \tilde{q}(x|y) \log \frac{p(x)}{\tilde{q}(x|y)} \mathrm{d}x. \tag{1.7}$$

If we only maximise the first term on the RHS, we tend to select $\tilde{q}$ who can explain the forward model well. If we only maximise the second term on the RHS, we tend to match the approximation with the prior distribution. In other words, the first term mimics the behaviour of the data fitting functional in Tikhonov regularisation, which defines the unregularised solution only matching the forward model. The second term mimics the behaviour of the regulariser, which reduces the ill-posedness of the problem and encodes special properties, e.g. smoothness and sparsity, into the solution. Thus the lower bound F can be regarded as a regularised functional.

1.2.1.2 Stochastic Variational Inference

Variational inference with KL divergence is finally transformed into the optimisation problem of an ELBO. However, conventional optimisation methods are not well scalable in large data settings for either mean field families or exponential families. Recent success of stochastic gradient descent (SGD), which enjoys both efficiency and accuracy, motivates the usage of stochastic optimisation methods for variational inference. Stochastic variational inference (SVI) [63] applies stochastic methods [117] to the optimisation of ELBOs by using natural gradients [7]. The usage of stochastic optimisation would only involve one example to evaluate the gradient of ELBO. Thus, the computational complexity is largely reduced. The usage of natural gradient is to find the direction along which the value of ELBO can be most efficiently decreased by leveraging the geometrical character.

SVI is developed for a class of conditionally conjugate models. The likelihood function is given by

$$p(y|x) = \prod_{i=1}^{n} p(y_i|x) \tag{1.8}$$

with each factor being an exponential family member

$$p(y_i|x) = h_1(y_i) \exp[x^t t(y_i) - A_1(x)]. \tag{1.9}$$

The prior distribution is required to be in the conjugate exponential family

$$p(x) = h_2(x)\exp[\alpha^t(x; -A_1(x)) - A_2(\alpha)], \tag{1.10}$$

where $\alpha = (\alpha_1; \alpha_2)$ (a MATLAB style of column vector stacking) being the natural parameter of $p(x)$. The distributions in the variational family (the same family as prior distributions) are written as

$$q(x) = h(x)\exp[\lambda^t(x; -A_1(x)) - A(\lambda)], \tag{1.11}$$

where $\lambda = (\lambda_1; \lambda_2)$ being the natural parameter of $q(x)$. Note that natural parameters are defined as the coefficients of the sufficient statistics in some exponential family and we refer to Appendix B for an example with Gaussian distributions.

Although SVI gives an efficient method for variational inference, it can not be applied to the Poisson models we are interested in. First of all, Poisson models with linear inverse link function do not belong to the exponential family of likelihood functions for SVI. Moreover, the restriction that the approximate distribution should be in the same conjugate family shrinks the spectrum of prior distributions we can choose from. For imaging tasks, we often adopt Gaussian type priors and Laplace type priors. Since Gaussian distributions are more reasonable approximate family, we at least cannot use Laplace type priors here. In fact, Gaussian type priors are not in the conjugate family of Poisson likelihood with exponential inverse link functions. To conclude, scalable variational inference for Poisson models is not straightforward and need further research.

1.2.1.3 Stein Variational Inference

Unlike stochastic variational inference, Stein variational inference [91] is a more general purpose method. It does not assume explicit forms of the likelihood function nor prior distribution barring the differentiability of the posterior distribution w.r.t. the unknown variable x. Apart from using much broader variational families, the key character of Stein variational inference is the particle update scheme rather than a conventional parameter optimisation. In this subsection, we will review Stein variational inference from the construction of the variational family to the implementation of the algorithm.

The first generalisation of Stein variational inference is the variational families with strong expressibility. The variational family $\mathscr{Q}$ is defined by a tractable reference $q_0(x)$ and a set of smooth one-to-one transforms T's. For a fixed T, the distribution of $x' = T(x)$ gives an element in $\mathscr{Q}$. In theory, such approximate family $\mathscr{Q}$ can give good approximation to almost arbitrary distributions [91]. For a fixed $q_0(x)$, the set of T's determines the expressibility of $\mathscr{Q}$. However, the choice of such set should be a balance among accuracy, tractability and solvability. As a result, [91] considers the set of T's with simple forms in some reproducing kernel Hilbert space (RKHS) $\mathscr{H}^d$. Specifically, each T is constructed by a perturbation scheme $T(x) = x + \varepsilon\phi(x)$, where $\phi(x) \in \mathscr{H}^d$ determines the

direction of perturbation and ε decides the step size. In stead of looking for an optimal T in one step, Stein variational inference iteratively finds some $T(x) = x + \varepsilon\phi(x)$ to reduce the KL divergence $\mathrm{KL}(q(x)||p(x|y))$, where $q(x)$ is the approximation to the true posterior distribution $p(x|y)$, and gives the optimal approximation when the process converges. This avoids explicitly using parametric forms of T nor calculating the Jacobian matrix of variable transformation [91]. It is shown in [91] that at the t-th iteration, the optimal $\phi_t^*(x)$ admits an explicit form. This motivates the particle update scheme idea that we can generate a set of samples $\{x_i^0\}_{i=1}^n$ from the reference distribution $q_0(x)$. At the t-th iteration, we can update the samples $\{x_i^{t-1}\}_{i=1}^n$ by $x_i^t = x_i^{t-1} + \varepsilon_t \phi_t^*(x_i^{t-1})$, where ε_t is the step size at the t-th iteration. When the algorithm converges, we can regard the updated particle $\{x_i\}_{i=1}^n$ as samples from the optimal approximate distribution $q(x)$.

In short, Stein variational inference updates the approximate posterior distribution by evolving the particles of it. And the evolution of the particles is based on a gradient-like scheme. A typical concern of gradient-like algorithms for inverse problems with Poisson data is the nonnegative constraint of the unobservable variable. For instance, in PET, the unobservable emission intensity is always nonnegative. Neglecting this property in gradient-like algorithms could lead to divergence. Although one can enforce the non-negativity by truncating the updated particle with lower bound zero after each iteration, this operation would introduce an extra evolution of the approximate distribution and invalidate the original interpretation of KL minimising. Moreover, typical priors adopted for imaging, e.g., total variation priors, are often non-smooth, which is not consistent with the differentiable assumption of Stein variational inference. One possible solution to solve this problem is adding a small number to smooth the total variation prior. However, it will introduce an extra hyperparameter which would increase the computational cost for hyperparameter tuning.

1.2.2 Expectation Propagation

Different from variational inference, expectation propagation (EP) is used for posterior distributions which admit a factorisation form and defines an iterative procedure to propagate the expectation information from each factor of the posterior distribution to the approximate distribution. Such propagation is achieved by an exclusion-inclusion scheme in every iteration. Preceding to the formal definition of the exclusion-inclusion scheme, we first define the factorisation of the posterior distribution $p(x|y)$.

Assume that the posterior distribution $p(x|y)$ admits a factorisation

$$p(x|y) = \prod_i t_i(x). \tag{1.12}$$

EP aims to find an approximation distribution $q(x)$ that admits the same factorisation

$$q(x) = \prod_i \tilde{t}_i(x). \tag{1.13}$$

Here each $\tilde{t}_i(x)$ is regarded as the approximation to the corresponding site function $t_i(x)$.

In each iteration, we first select a site function $t_i(x)$ to approximate. Then we exclude the i-th site approximation $\tilde{t}_i(x)$ from the approximation distribution $q(x)$ and form a cavity distribution

$$q_{\setminus i}(x) \propto \prod_{j \neq i} \tilde{t}_j(x). \tag{1.14}$$

Then we include the i-th site function $t_i(x)$ into the cavity distribution and form a tilted distribution

$$\hat{q}(x) \propto t_i(x) q_{\setminus i}(x). \tag{1.15}$$

Then we match the moments of $q(x)$ to those of $\hat{q}(x)$. The i-th approximation site function $\tilde{t}_i(x)$ is updated such that

$$q(x) = \tilde{t}_i(x) \prod_{j \neq i} \tilde{t}_j(x). \tag{1.16}$$

The moment matching step can be interpreted as minimising the KL divergence in an opposite direction as variational inference, i.e.

$$q(x) = \arg \min_{\tilde{q}(x) \in \mathcal{Q}} \mathrm{KL}(\hat{q}(x) || \tilde{q}(x)). \tag{1.17}$$

Due to the independent and identically distributed (i.i.d.) assumption of the data, the likelihood functions admit natural factorisations. Since we normally use Gaussian approximations, Gaussian prior can be directly updated without factorisation approximations. As for other prior distributions, the usage of EP might not be so straightforward. For example, while anisotropic total variation priors admit factorisation forms to which EP can be applied, isotropic total variation priors do not enjoy such property.

Despite the widely successful applications, e.g. Gaussian processes [112, 23], electrical impedance tomography [52], etc., theoretical understanding of EP is quite limited. We refer interested readers to recent results [35, 34] upon this regard. Beside the lack of theoretical analysis, sometimes EP can suffer from numerical instability. To alleviate the instability, there are mainly two solutions, i.e. improving the accuracy of integral evaluations of expectations and use fractional exclusion-inclusion schemes. While the first solution will be addressed in our second work, we will sketch the idea of fractional update here. For the propagation of the i-th factor, instead of taking out the whole factor $t_i(x)$, one can only take a fraction of it out, say $t_i(x)^{1/n}$, and include the same fraction of approximation into the approximation, i.e. $\tilde{t}_i(x)^{1/n}$. Similar to the interpretation of minimising KL divergence for whole exclusion-inclusion scheme, doing fractional update can be regarded as minimising some α-divergence [97]. In the literature, the EP with fractional update is often referred to as power EP [97] or fractional EP [53].

Note that the expectation evaluations of EP normally involves integrations in a very high di-

mensional space. Although the projection form in many practical models can render this issue into lower dimensional integrals, the high computational complexity still calls for further careful treatment. Besides the computational efficiency, the memory efficiency is also a problem for big data settings. Stochastic expectation propagation (SEP) [87] addresses this issue by only maintaining global parameters and thus saves space for local approximation parameters.

1.2.3 Bayesian Deep Learning

In recent decade, the success of deep learning [86] evokes a revolution in the machine learning community. Due to the strong approximation ability of deep neural networks, they perform much better than conventional methods in many applications, e.g., computer vision (CV), natural language processing (NLP), reinforcement learning (RL), etc. While Bayesian inference and deep learning have been two parallel research areas, the combination of them can actually date back to the last century [94].

Neural networks can be used to learn an inverse model f from data. However, what neural networks give is only a point estimate $x^* = f(y^*)$ for an input y^*. Thus, the important information of uncertainty quantification is not available. With the Bayesian framework, we can actually enable uncertainty quantifications of neural networks. [94] proposed a Bayesian framework for the weights of the neural networks. The likelihood functions are constructed based on errors for the data and prior distributions are constructed based on the regularisation need. Following this route, [48] shows that doing dropout training can be interpreted as conducting variational inference in deep Gaussian processes. Dropout is essentially one kind of perturbation on the parameters w of neural networks, which introduces randomness into the variable w. With the approximate posterior distribution of w, the uncertainty of the network output is propagated from the randomness of the network parameter w for a fixed input-output pair. Instead of doing dropout training, [79] proposed a Gaussian perturbation scheme to Adam optimiser and interpreted it as a natural gradient algorithm to do fast Gaussian mean-field variational inference. The above mentioned methods conduct Bayesian inference on neural networks parameters. In doing so, they focus on the *epistemic* uncertainty, a.k.a. model uncertainty, which is about possible models fitting the data. However, in our problem settings, we are interested in the uncertainty of the unobservable variable that could lead to the observation with the forward process and prior information, which belongs to another kind of uncertainty, i.e., *aleatoric* uncertainty, and is the uncertainty of the data itself. Thus, the direction of studying neural networks with Bayesian inference on weights deviates from the scope our research.

To enable the *aleatoric* uncertainty quantification with neural networks, one idea is to let neural networks output the parameters of some distribution family of the unobservable variable [102, 85, 58, 50]. The networks are trained to fit the approximate distributions to the true posterior distribution, which is consistent with the interpretation of *aleatoric* uncertainty. For example, for Gaussian distributions, such neural networks are expected to output the mean and variance [102].

One can also use an ensemble of networks to output the parameters of a mixture of distributions [85].

Besides only outputting a distribution at the last layer, [50] actually extends this idea to every layer of the network. Then the forward propagation of the probabilistic neural net is defined by the assumed density filtering (ADF) algorithm which is a simplified version of expectation propagation (EP). The extension from conventional neural nets to the probabilistic neural nets is a lightweight way as argued in [50], i.e., without much extra burden on number of parameters and computations. The improvement of this extension is that it enables the query of uncertainty on each layer of the network, while it is not clear how it additionally benefits uncertainty quantification for the final output of interest.

Although [85] allows for training a mixture of finite many distributions, which enlarges the distribution families a single network can parameterise, it might be still insufficient to model complex distributions. Conditional variational auto-encoders (CVAEs) [126] introduced an intermediate variable z and model the target distribution by $p(x|y) = \int p(x|z,y)p(z|y)\mathrm{d}z$. z acts as a mixture variable and extends the mixture of finite many distributions to the mixture of infinite many distributions. As a conditional variant of variational auto-encoders (VAEs), CVAEs also model the target distribution by transforming samples from a simple distribution, i.e., mean field Gaussian distributions, into samples from the target distribution. However, the objective function in CVAEs is not direct conditional version of that of VAEs. In other words, one can not simply interpret the distribution given by CVAEs as approximation distribution from variational inference with KL divergence. Hence, further theoretical understanding is needed to justify the application of CVAEs to inverse problems.

Apart from the conditional variant of VAEs, recent advances also investigate conditional generative adversarial nets (Conditional GANs) for inverse problems [4]. The objective function of Conditional GANs is equivalent to Wasserstein 1 distance under some technical assumption, i.e., Lipschitz condition, which is not straightforward to strictly enforce for neural networks. In the literature, practitioners either softly incorporate the penalty on gradients [57] or clip the gradients with some threshold [12]. It is worth noting that unconditional versions of the generative modelling algorithms, i.e., VAEs and GANs, are unsupervised algorithms. In other words, one cannot incorporate ground truth data into these algorithms, despite that for highly ill-posed inverse problems, information encoded in observations only may not be sufficient for data-driven modelling.

Before concluding the literature review, we shall compare the deep learning based full Bayesian approaches with conventional full Bayesian approaches, e.g., variational inference, expectation propagation, etc. Abstractly speaking, all full Bayesian approaches provide a distribution of the unobservable variable x for a given observation y, as an approximation to the true posterior distribution $p(x|y)$. The difference between learning based full Bayesian approaches and conventional ones is that the whole procedure of conventional approaches needs to be rerun for every new observation. For sampling methods, new chains of sampling shall be simulated. For variational inference, the optimisation problems with respect to the parameter of the variational families need to be solved.

For expectation propagation, the moment evaluation procedures need to be conducted for all site functions. All of above repetitions would greatly increase the computational cost and undermine the efficiency of the algorithm in production scenarios. In contrast, for deep learning based algorithms, once the neural networks are trained, for each new observation y, the procedure deriving the approximation distribution is deduced to the feedforward process, which is composed of simple linear transforms and simple non-linearities and thus very efficient on modern computational equipments, e.g., GPUs and TPUs. Although the feedforward process of DNNs is very efficient, the offline training of them usually needs large amount of time. To further improve the efficiency and scalability of deep learning based full Bayesian approaches, attention need to be paid on how to reduce the training time, which will be addressed in our third work.

1.2.4 Conclusion

With discussions above, we can see that designing scalable approximate inference methods for inverse problems with Poisson data is not straightforward and need sophisticated investigations. In the literature of statistical/machine learning, inverse problems with Poisson data are mostly equally treated as other generalised linear models (GLMs). Thus, specific characteristics, e.g., structures or constraints, which are important factors for scalable, accurate and flexible inversions, are not always taken into consideration. In this part, we shall stress this point with three practical and important characteristics for inverse problems with Poisson data.

The first characteristic is the low rank structure of forward operators. For instance, in X-ray CT, the forward operator is the discretised Radon matrix $A \in \mathbb{R}^{n\times m}$. While m is the number of pixels of the image, n is the number of observations which is given by the product of number of beams per angle and number of angles. Fix other parameters, the fewer the number of angles is or the fewer the number of beams is, the fewer information we will get from the observation. As a result, the more difficult the problem will be. This difficulty is reflected by the ill-conditionedness of the matrix A. A key character of ill-conditioned matrices is the fast singular value decay. By Eckart Young Mirsky theorem, a matrix with fast singular value decay, which is referred to as a matrix with low rank structure, can be well-approximated by low rank matrices. This structure is also quite common for other ill-posed inverse problems defined by first kind of Fredholm integrals and has been shown great potential to accelerate algorithms solving inverse problems in the deterministic setting [137]. Thus, it is very interesting to investigate how it can be leveraged to accelerate full Bayesian approaches, which suffer from poor scalability of dimensions.

The second characteristic is the non-negativity of the unobservable variable. For example, in PET, the variable of interest is the intensity of photon emissions, which is naturally equal to or greater than zero. Neglecting such constraint would no doubt leads to unrealistic reconstructions. Algorithms for point estimates, e.g., MLE, are usually formulated as optimisation problems with x being the variable for optimisation. It is straightforward to incorporate the nonnegative constraint

on x to these algorithms by casting the optimisation problems as constrained optimisation problems. Although in some full Bayesian approaches, the algorithms would also admit some variational problem, e.g., variational inference, the variable for optimisation is the distribution of x, e.g., $q(x) \in \mathcal{Q}$, which generally do not accept constraint incorporations. Hence, it is of great interest to investigate how to incorporate the nonnegative constraint of unobservable x into the process of full Bayesian approaches.

The third characteristic is the presence of forward operators in inverse problems and their importance for deep learning based approaches. The forward operators are often given by fundamental physical laws, e.g., Radon matrices from the Beer-Lambert law, and serve as a part of established prior knowledge. The presence of forward operators is a unique feature in inverse problems and is not common in the machine learning setting. Such knowledge is also encoded in the unobservable and observable data pairs $\{x_i, y_i\}_i$ needed for supervised learning, but in an implicit way. Since neural networks extract features in a black box manner, one cannot make sure that the physical laws are respected by neural networks, which otherwise may lead to overfitting the correspondence relations in training data. Besides, the forward operators and their adjoints connect the space of unobservable variable and the space of the observable variable, which are usually of different dimensions. Disentangling the unobservable space and the observable space by the operators would be beneficial to prevent overfitting special features only emerging in the observable space. To conclude, it is very important to investigate how to incorporate the forward operators and their adjoints into deep learning based full Bayesian approaches.

1.3 Overview and Contribution

After reviewing inverse problems with Poisson data and full Bayesian approaches for posterior explorations, we conclude that many theoretical and practical concerns should be addressed for scalable and accurate deployment. In this thesis, we would address the discussed concerns and try to shed a light on relevant issues. Before detailed discussions, we will highlight the roadmap and contributions of our work in this section.

In Chapter 2, we analyse variational inference with multivariate Gaussian approximations to the posterior distribution arising from the Poisson model with a Gaussian prior. This is achieved by seeking an optimal Gaussian distribution minimising the Kullback Leibler divergence from the posterior distribution to the approximation, or equivalently maximising the lower bound for the model evidence. We derive an explicit expression for the lower bound, and show the existence and uniqueness of the optimal Gaussian approximation. The lower bound functional can be viewed as a variant of classical Tikhonov regularisation that penalises also the covariance. Then we develop an efficient alternating direction maximisation algorithm for solving the optimisation problem, and analyse its convergence. We discuss strategies for reducing the computational complexity via low rank structure of the forward operator and the sparsity of the covariance. Further, as an application of the lower

bound, we discuss hierarchical Bayesian modelling for selecting the hyperparameter in the prior distribution, and propose a monotonically convergent algorithm for determining the hyperparameter. We present numerical experiments to illustrate the Gaussian approximation and the algorithms. Note that this chapter is based on the published paper *Variational Gaussian Approximation for Poisson Data* [14].

In Chapter 3, we develop an approximate Bayesian inference technique based on expectation propagation for approximating the posterior distribution formed from the Poisson likelihood function and a Laplace type prior distribution, e.g. the anisotropic total variation prior. The approach iteratively yields a Gaussian approximation, and at each iteration, it updates the Gaussian approximation to one factor of the posterior distribution by moment matching. We derive explicit update formulae in terms of one-dimensional integrals, and also discuss stable and efficient quadrature rules for evaluating these integrals. The method is showcased on two-dimensional PET images. Note that this chapter is based on the published paper *Expectation Propagation for Poisson Data* [140].

In Chapter 4, we develop a novel computational framework, termed as Probabilistic Iterative Networks (PIN), to output a distribution of the unobservable variable(s) that approximates the posterior distribution for each observation. The framework is very general and flexible: It can handle implicit noise models and priors, can incorporate physically important forward maps and their adjoints, and is transferable between different datasets (e.g., input/output of different dimensions). Once the network is trained, it provides an efficient sampler for an approximate posterior distribution via feedforward propagation, and the summarising statistics of the generated samples can be used for both point estimation and uncertainty quantification. We illustrate the proposed framework with numerical experiments on PET, and the numerical results show that the samples are of high quality when compared with state-of-the-art benchmark methods. Note that this chapter is based on the working paper *Probabilistic Residual Learning for Aleatoric Uncertainty in Image Restoration* [141] and the working paper *Probabilistic Iterative Networks for Inverse Problems*.

In Chapter 5, we conclude the project and discuss future research regimes.

Chapter 2

Variational Gaussian Approximation for Poisson Data

2.1 Introduction

In the previous chapter, we discuss the motivation of our research on approximate inference for Poisson data and concluded that applications of approximate inference methods should take specific characters of concrete problems into consideration. In this chapter, we shall focus on the case of a Gaussian prior, which forms the basis of many other important priors, e.g., sparsity prior via scale mixture representation. Then following the Bayesian procedure, we arrive at a posterior probability distribution, which however is analytically intractable due to the nonstandard form of the likelihood function for the Poisson model. We will explain this more precisely in Section 2.2. To explore the posterior state space, instead of applying popular general-purposed sampling techniques, e.g., Markov chain Monte Carlo (MCMC), we employ a variational Gaussian approximation (VGA). The VGA is one extremely popular approximate inference technique in machine learning [134, 26]. Specifically, we seek an optimal Gaussian approximation to the non-Gaussian posterior distribution with respect to the Kullback-Leibler divergence. The approach leads to a large-scale optimisation problem over the mean $\bar{x}$ and covariance C (of the Gaussian approximation). In practice, it generally delivers an accurate approximation in an efficient manner, and thus has received immense attention in recent years in many different areas [62, 15, 26, 11]. By its very construction, it also gives a lower bound to the model evidence, which facilitates its use in model selection. However, a systematic theoretical understanding of the approach remains largely missing.

In this work, we shall study the analytical properties and develop an efficient algorithm for the VGA in the context of Poisson data (with the exponential inverse link function). We shall provide a detailed analysis of the resulting optimisation problem. The study sheds interesting new insights into the approach from the perspective of regularisation. Our main contributions are as follows. First, we derive explicit expressions for the objective functional and its gradient, and establish its strict concavity and the well-posedness of the optimisation problem. Second, we develop an efficient

numerical algorithm for finding the optimal Gaussian approximation, and discuss its convergence properties. The algorithm is of alternating maximisation (coordinate ascent) nature, and it updates the mean $\bar{x}$ and covariance C alternatingly by a globally convergent Newton method and a fixed point iteration, respectively. We also discuss strategies for its efficient implementation, by leveraging realistic structure of inverse problems, e.g., low-rank nature of the forward map A and sparsity of the covariance C, to reduce the computational complexity. Third, we illustrate the use of the evidence lower bound for hyperparameter selection within an empirical Bayesian framework, leading to a purely data-driven approach for determining the regularisation parameter, whose proper choice is notoriously challenging. We shall develop a monotonically convergent algorithm for determining the hyperparameter in the Gaussian prior. Last, we illustrate the approach and the algorithms with numerical experiments for one- and two-dimensional examples.

Last, we discuss existing works on Poisson models. The majority of existing works aim at recovering point estimators, either iteratively or by a variational framework [64]. Recently, Bardsley and Luttman [16] described a Metroplis-Hastings algorithm for exploring the posterior distribution (with rectified linear inverse link function), where the proposal samples are drawn from the Laplace approximation (cf. Remark 2.3.1). The Poisson model (2.2) belongs to generalised linear models (GLMs), to which the VGA has been applied in statistics and machine learning [103, 78, 26, 120]. Ormerod and Wand [103] suggested a variational approximation strategy for fitting GLMs suitable for grouped data. Challis and Barber [26] systematically studied VGA for GLMs and various extensions. The focus of these interesting works [103, 78, 26, 120] is on the development of the general VGA methodology and its applications to concrete problems, and do not study analytical properties and computational techniques for the lower bound functional, which is the main goal of this work.

The rest of the chapter is organised as follows. In Section 2.2, we describe the Poisson model, and formulate the posterior probability distribution. Then in Section 2.3, we develop the variational Gaussian approximation, and analyse its basic analytical properties. In Section 2.4, we propose an efficient numerical algorithm for finding the optimal Gaussian approximation, and in Section 2.5, we apply the lower bound to hyperparameter selection within an empirical Bayesian framework. In Section 2.6 we present numerical results for several examples. In Appendix A.1 and Appendix A.2, we provide further discussions on the convergence of the fixed point iteration (2.12) and the differentiability of the regularised solution.

2.2 Notation and Problem Setting

First we recall some standard notation in linear algebra. Throughout, (real-valued) vectors and matrices are denoted by lower- and upper-case letters, respectively, and the vectors are always column vectors. We will use the notation $(\cdot,\cdot)$ to denote the usual Euclidean inner product. We shall slightly abuse the notation $(\cdot,\cdot)$ also for the inner product for matrices. That is, for two matrices $X, Y \in \mathbb{R}^{n\times m}$,

we define

$$(X,Y) = \mathrm{tr}(XY^t) = \mathrm{tr}(X^tY),$$

where $\mathrm{tr}(\cdot)$ denotes taking the trace of a square matrix, and the superscript t denotes the transpose of a vector or matrix. This inner product induces the usual Frobenius norm for matrices. We shall use extensively the cyclic property of the trace operator $\mathrm{tr}(\cdot)$: for three matrices X,Y,Z of appropriate size, there holds

$$\mathrm{tr}(XYZ) = \mathrm{tr}(YZX) = \mathrm{tr}(ZXY).$$

We shall also use the notation $\mathtt{diag}(\cdot)$ for a vector and a square matrix, which gives a diagonal matrix and a column vector from the diagonals of the matrix, respectively, in the same manner as the `diag` function in `MATLAB`. The notation $\mathbb{N} = \{0,1,\ldots\}$ denotes the set of natural numbers. Further, the notation $\circ$ denotes the Hadamard product of two matrices or vectors. Last, we denote by $\mathscr{S}_m^+ \subset \mathbb{R}^{m\times m}$ the set of symmetric positive definite matrices in $\mathbb{R}^{m\times m}$, I_m the identity matrix in $\mathbb{R}^{m\times m}$, and by $|\cdot|$ and $\|\cdot\|$ the determinant and the spectral norm, respectively, of a square matrix. Throughout, we view exponential, logarithm and factorial of a vector as componentwise operation.

Next we recall the finite-dimensional Poisson data model. Let $x \in \mathbb{R}^m$ be the unknown signal, $a_i \in \mathbb{R}^m$, $i = 1,\ldots,n$, and $y \in \mathbb{N}^n \subset \mathbb{R}^n$ be the data vector. We stack the column vectors a_i into a matrix A by $A = [a_i^t] \in \mathbb{R}^{n\times m}$. Given the matrix A and data $y \in \mathbb{N}^n$, the Poisson model takes the form:

$$y_i \sim \mathrm{Pois}(e^{(a_i,x)}), \quad i = 1,2,\ldots,n.$$

Thus, the likelihood function $p(y_i|x)$ for the data point y_i is given by

$$p(y_i|x) = \frac{\lambda_i^{y_i} e^{-\lambda_i}}{y_i!}, \quad \lambda_i = e^{(a_i,x)},\ i = 1,\ldots,n. \tag{2.1}$$

It is worth noting that the exponential function enters into the Poisson parameter λ. This is commonly known as the log link function or log-linear model in the statistical literature [25]. There are several other models for the (inverse) link functions, e.g., rectified-linear and softplus [109], each having its own pros and cons for modelling count data. In this work, we shall focus on the log link function. Also this model can be viewed as a simplified statistical model for transmission tomography [139, 43].

The likelihood function $p(y_i|x)$ can be equivalently written as

$$p(y_i|x) = e^{y_i(a_i,x) - e^{(a_i,x)} - \ln(y_i!)}.$$

Under the independent identically distributed (i.i.d.) assumption on the data points y_i, the likelihood

function $p(y|x)$ of the data vector y is given by

$$p(y|x) = \prod_{i=1}^{n} p(y_i|x) = e^{(Ax,y)-(e^{Ax},1_n)-(\ln(y!),1_n)}, \tag{2.2}$$

where $1_n \in \mathbb{R}^n$ is the vector with all entries equal to unity, i.e., $1_n = [1,\dots,1]^t \in \mathbb{R}^n$.

Further, we assume that the unknown x follows a Gaussian prior $p(x)$, i.e.,

$$p(x) = p_{\mathcal{N}}(x;\mu_0,C_0) := (2\pi)^{-\frac{m}{2}}|C_0|^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\mu_0)^t C_0^{-1}(x-\mu_0)},$$

where $\mu_0 \in \mathbb{R}^m$ and $C_0 \in \mathcal{S}_m^+$ denote the mean and covariance of the Gaussian prior, respectively, and $\mathcal{N}$ denotes the normal distribution. In the framework of variational regularisation, the corresponding penalty $\frac{1}{2}(x-\mu_0)^t C_0^{-1}(x-\mu_0)$ often imposes certain smoothness constraint. The Gaussian prior $p(x)$ may depend on additional hyperparameters, cf. Section 2.5 for details. Then by Bayes' formula, the posterior probability distribution $p(x|y)$ is given by

$$p(x|y) = Z^{-1}(y)p(x,y), \tag{2.3}$$

where the joint distribution $p(x,y)$ is defined by

$$p(x,y) = (2\pi)^{-\frac{m}{2}}|C_0|^{-\frac{1}{2}}e^{(Ax,y)-(e^{Ax},1_n)-(\ln(y!),1_n)-\frac{1}{2}(x-\mu_0)^t C_0^{-1}(x-\mu_0)},$$

and the normalising constant $Z(y)$, which depends only on the given data y, is given by

$$Z(y) = p(y) = \int p(x,y)\mathrm{d}x.$$

That is, the normalising constant $Z(y)$ is an integral living in a very high-dimensional space if the parameter dimension m is large. Thus it is computationally intractable, and so is the posterior distribution $p(x|y)$, since it also involves the constant $Z(y)$.

The posterior distribution $p(x|y)$ given in (2.3) is the Bayesian solution to the Poisson model (2.1) (under a Gaussian prior), and it contains all the information about the inverse problem. In this work, we shall employ the VGA to obtain an optimal Gaussian approximation $q(x)$ to the posterior distribution $p(x|y)$ in the Kullback-Leibler divergence $\mathrm{D_{KL}}(q||p)$. Fitting a Gaussian to an intractable distribution is a well-adopted choice for approximate Bayesian Inference, and it has demonstrated success in many practical applications [62, 15, 26, 11]. The popularity can be largely attributed to the fact that the Gaussian approximation is computationally attractive due to the good analytical properties, and yet delivers reasonable accuracy for a wide range of problems. However, analytical properties of Approximate Inference procedures are rarely studied. In the context of Poisson mixed models, the asymptotic normality of the estimator and its convergence rate was analysed [60]. In a

general setting, some theoretical issues were studied in [110, 92].

2.3 Gaussian Variational Approximation

In this section, we derive explicit expressions for the lower bound functional and its gradient, and discuss basic analytic properties, e.g., concavity and existence.

Remark 2.3.1. *In practice, the so-called Laplace approximation is quite popular [129]. Specifically, let $\hat{x}$ be the maximum a posteriori (MAP) estimator $\hat{x}$, i.e., $\hat{x} = \arg\min_{x\in\mathbb{R}^m} g(x)$, where $g(x) = -\ln p(x|y)$ is the negative log posterior distribution. Consider the Taylor expansion of $g(x)$ at the MAP estimator $\hat{x}$:*

$$\begin{aligned} g(x) &\approx g(\hat{x}) + (\nabla g(\hat{x}), x-\hat{x}) + \tfrac{1}{2}(x-\hat{x})^t H(x-\hat{x}) \\ &= g(\hat{x}) + \tfrac{1}{2}(x-\hat{x})^t H(x-\hat{x}), \end{aligned}$$

since $\nabla g(\hat{x})$ vanishes. The Hessian H of $g(x)$ is given by

$$H = A^t \mathrm{diag}(e^{A\hat{x}})A + C_0^{-1}.$$

Thus, $\hat{x}$ might serve as an approximate posterior mean, and the inverse Hessian H^{-1} as an approximate posterior covariance. However, unlike the VGA discussed below, it lacks the optimality as evidence lower bound (within the Gaussian family), and thus may be suboptimal for model selection etc.

2.3.1 Variational Gaussian Lower Bound

By substituting $p(x)$ with the posterior distribution $p(x|y)$ in Equation (1.1), we obtain

$$\mathrm{D_{KL}}(q(x|y)||p(x|y)) = \int q(x|y)\ln\frac{q(x|y)}{p(x|y)}\mathrm{d}x.$$

Since the posterior distribution $p(x|y)$ depends on the unknown normalising constant $Z(y)$, the integral on the right hand side is not computable. Nonetheless, given y, $Z(y)$ is fixed. In view of the identity

$$\ln Z(y) = \int q(x|y)\ln\frac{p(x,y)}{q(x|y)}\mathrm{d}x + \int q(x|y)\ln\frac{q(x|y)}{p(x|y)}\mathrm{d}x,$$

instead of minimising $\mathrm{D_{KL}}(q(x|y)||p(x|y))$, we may equivalently maximise the functional

$$F(q,y) = \int q(x|y)\ln\frac{p(x,y)}{q(x|y)}\mathrm{d}x. \tag{2.4}$$

$F(q,y)$ provides a lower bound on the model evidence $Z(y)$, for any choice of the distribution q. For any fixed q, $F(q,y)$ may be used as a substitute for the analytically intractable model evidence $Z(y)$, and hence it is called an evidence lower bound (ELBO). Since the data y is fixed, it will be suppressed from $F(q,y)$ below. In the VGA, we restrict our choice of q to Gaussian distributions.

Meanwhile, a Gaussian distribution $q(x)$ is fully characterised by its mean $\bar{x} \in \mathbb{R}^m$ and covariance $C \in \mathscr{S}_m^+ \subset \mathbb{R}^{m\times m}$, i.e.,

$$q(x) = p_{\mathscr{N}}(x;\bar{x},C).$$

Thus, $F(q)$ is actually a function of $\bar{x} \in \mathbb{R}^m$ and $C \in \mathscr{S}_m^+$, and will be written as $F(\bar{x},C)$ below. Then the approach seeks optimal variational parameters $(\bar{x},C)$ to maximise ELBO. This step turns a challenging sampling problem into a computationally more tractable optimisation problem.

The next result gives an explicit expression for the lower bound $F(\bar{x},C)$.

Proposition 2.3.1. *For any fixed y, μ_0 and C_0, the lower bound $F(\bar{x},C)$ is given by*

$$\begin{aligned} F(\bar{x},C) = (y,A\bar{x}) - (1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)}) - \tfrac{1}{2}(\bar{x}-\mu_0)^t C_0^{-1}(\bar{x}-\mu_0) - \tfrac{1}{2}\mathrm{tr}(C_0^{-1}C) \\ + \tfrac{1}{2}\ln|C| - \tfrac{1}{2}\ln|C_0| + \tfrac{m}{2} - (1_n, \ln(y!)). \end{aligned} \tag{2.5}$$

Proof. By the definition of the functional $F(\bar{x},C)$ and the joint distribution $p(x,y)$, we have

$$\begin{aligned} F(\bar{x},C) = \int p_{\mathscr{N}}(x;\bar{x},C)\Big[\ln|C_0|^{-\frac{1}{2}} - \ln|C|^{-\frac{1}{2}} + (Ax,y) - (e^{Ax},1_n) - (\ln(y!),1_n) \\ - \tfrac{1}{2}(x-\mu_0)^t C_0^{-1}(x-\mu_0) + \tfrac{1}{2}(x-\bar{x})^t C^{-1}(x-\bar{x})\Big]\mathrm{d}x. \end{aligned}$$

It suffices to evaluate the integrals termwise. Clearly, we have $\int p_{\mathscr{N}}(x;\bar{x},C)(Ax,y)\mathrm{d}x = (A\bar{x},y)$. Next, using moment generating function, we have

$$\begin{aligned} \int p_{\mathscr{N}}(x;\bar{x},C)(e^{Ax},1_n)\mathrm{d}x &= \sum_i \int p_{\mathscr{N}}(x;\bar{x},C)e^{(a_i,x)}\mathrm{d}x \\ &= \sum_i e^{(a_i,\bar{x})+\frac{1}{2}a_i^t C a_i} = (1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)}). \end{aligned}$$

With the Cholesky decomposition $C = LL^t$, for $z \sim \mathscr{N}(0,I_m)$, $x = \mu + Lz \sim \mathscr{N}(x;\mu,C)$. This and the bias-variance decomposition yield ($\mathbb{E}_{q(x)}[\cdot]$ takes expectation with respect to the density $q(x)$): for any symmetric $X \in \mathbb{R}^{m\times m}$

$$\mathbb{E}_{q(x)}[x^t X x] = \mathbb{E}_{\mathscr{N}(z;0,I_m)}[(\mu+Lz)^t X(\mu+Lz)] = \mu^t X\mu + \mathbb{E}_{\mathscr{N}(z;0,I_m)}[z^t L^t X L z].$$

By the cyclic property of trace, we have $\mathbb{E}_{\mathscr{N}(z;0,I_m)}[z^t L^t X L z] = \mathrm{tr}(L^t X L) = \mathrm{tr}(XLL^t) = \mathrm{tr}(XC)$. In particular, this gives

$$\mathbb{E}_{q(x)}[(x-\mu_0)^t C_0^{-1}(x-\mu_0)] = (\bar{x}-\mu_0)^t C_0^{-1}(\bar{x}-\mu_0) + \mathrm{tr}(C_0^{-1}C),$$

and

$$\mathbb{E}_{q(x)}[(x-\bar{x})^t C^{-1}(x-\bar{x})] = m.$$

Collecting preceding identities completes the proof of the proposition. □

Remark 2.3.2. *The terms in the functional $F(\bar{x},C)$ in* (2.5) *admit interesting interpretation in the lens of classical Tikhonov regularisation* (*see, e.g., [42, 67, 123]*). *To this end, we rewrite it as*

$$\begin{aligned} F(\bar{x},C) =& (y,A\bar{x}) - (1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)}) - (1_n, \ln(y!)) \\ & - \tfrac{1}{2}(\bar{x}-\mu_0)^t C_0^{-1}(\bar{x}-\mu_0) \\ & - \tfrac{1}{2}\mathrm{tr}(C_0^{-1}C) + \tfrac{1}{2}\ln|C| - \tfrac{1}{2}\ln|C_0| + \tfrac{m}{2}. \end{aligned}$$

The first line represents the fidelity or "pseudo-likelihood" function. It is worth noting that it actually involves the covariance C. In the absence of the covariance C, it recovers the familiar log likelihood for Poisson data, cf. Remark 2.3.1. The second line imposes a quadratic penalty on the mean $\bar{x}$. This term recovers the familiar penalty in Tikhonov regularisation (*except that it is imposed on $\bar{x}$*). *Recall that the function $-\ln|C|$ is strictly convex in $C \in \mathscr{S}_m^+$ [49, Lemma 6.2.2]. Thus, one may define the corresponding Bregman divergence $d(C,C_0)$. In view of the identities [39]*

$$\frac{\partial}{\partial C}\mathrm{tr}(CC_0^{-1}) = C_0^{-1} \quad \textit{and} \quad \frac{\partial}{\partial C}\ln|C| = C^{-1} \tag{2.6}$$

simple computation gives the following expression for the divergence:

$$d(C,C_0) = \mathrm{tr}(C_0^{-1}C) - \ln|C_0^{-1}C| - m \geq 0.$$

Statistically, it is the Kullback-Leibler divergence between two Gaussians of identical mean. The divergence $d(C,C_0)$ provides a distance measure between the prior covariance C_0 and the posterior one C. Let $\{(\lambda_i, v_i)\}_{i=1}^m$ be the pairs of generalised eigenvalues and eigenfunctions of the pencil (C,C_0), i.e., $Cv_i = \lambda_i C_0 v_i$. Then it can be expressed as

$$d(C,C_0) = \sum_{i=1}^{m} (\lambda_i - \ln\lambda_i - 1).$$

This identity directly indicates that $d(C,C_0) \leq c$ implies $\|C\| \leq c$ and $\|C^{-1}\| \leq c$, where here and below c denotes a generic constant which may change at each occurrence.

Thus, the third line regularises the posterior covariance C by requesting nearness to the prior one C_0 in Bregman divergence. It is interesting to observe that the Gaussian prior implicitly induces a penalty on C, although it is not directly enforced. In statistics, the Bregman divergence $d(C,C_0)$ is also known as Stein's loss [69].

2.3.2 Theoretical Properties of the Lower Bound

Now we study basic analytical properties, i.e., concavity, existence and uniqueness of maximiser, and gradient of the functional $F(\bar{x},C)$ defined in (2.5), from the perspective of optimisation.

A first result shows the concavity of $F(\bar{x},C)$. Let X and Y be two convex sets. Recall that a functional $f : X \times Y \to \mathbb{R}$ is said to be jointly concave, if and only if

$$f(\lambda x_1 + (1-\lambda)x_2, \lambda y_1 + (1-\lambda)y_2) \geq \lambda f(x_1,y_1) + (1-\lambda) f(x_2,y_2)$$

for all $x_1, x_2 \in X$, $y_1, y_2 \in Y$ and $\lambda \in [0,1]$. Further, f is called strictly jointly concave if the inequality is strict for any $(x_1,y_1) \neq (x_1,y_1)$ and $\lambda \in (0,1)$. It is easy to see that $\mathscr{S}_m^+$ is a convex set.

Theorem 2.3.1. *For any $C_0 \in \mathscr{S}_m^+$, the functional $F(\bar{x},C)$ is strictly jointly concave with respect to $\bar{x} \in \mathbb{R}^m$ and $C \in \mathscr{S}_m^+$.*

Proof. It suffices to consider the terms apart from the linear terms $(y, A\bar{x})$ and $-\frac{1}{2}\mathrm{tr}(C_0^{-1}C)$ and the constant term $-\frac{1}{2}\ln|C_0| + \frac{m}{2} - (1_n, \ln(y!))$. Since $A\bar{x} + \frac{1}{2}\mathrm{diag}(ACA^t)$ is linear in $\bar{x}$ and C, and exponentiation preserves convexity, the term $-(1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})$ is also jointly concave. Next, the term $-\frac{1}{2}(\bar{x}-\mu_0)^t C_0^{-1}(\bar{x}-\mu_0)$ is strictly concave for any $C_0 \in \mathscr{S}_m^+$. Last, the log-determinant $\ln|C|$ is strictly concave over $\mathscr{S}_m^+$ [49, Lemma 6.2.2]. The assertion follows since strict concavity is preserved under summation. □

Next, we show the well-posedness of the optimisation problem in VGA.

Theorem 2.3.2. *There exists a unique pair of $(\bar{x},C)$ solving the optimisation problem*

$$\max_{\bar{x}\in\mathbb{R}^m, C\in\mathscr{S}_m^+} F(\bar{x},C) \tag{2.7}$$

Proof. The proof follows by direct methods in calculus of variation, and we only briefly sketch it. Clearly, there exists a maximising sequence, denoted by $\{(\bar{x}^k,C^k)\} \subset \mathbb{R}^m \times \mathscr{S}_m^+$, and we may assume $F(\bar{x}^k,C^k) \geq c =: F(\mu_0,C_0)$. Thus, by (2.5) in Proposition 2.3.1 and the divergence $d(C,C_0)$, we have

$$(A\bar{x}^k, y) - (\bar{x}^k-\mu_0)^t C_0^{-1}(\bar{x}^k-\mu_0) - d(C^k,C_0) \geq c + (e^{A\bar{x}^k+\frac{1}{2}\mathrm{diag}(AC^kA^t)}, 1_n) \geq c.$$

By the Cauchy-Schwarz inequality, we have $(\bar{x}^k-\mu_0)^t C_0^{-1}(\bar{x}^k-\mu_0) + d(C^k,C_0) \leq c$. This immediately implies a uniform bound on $\{(\bar{x}^k,C^k)\}$ and $\{(C^k)^{-1}\}$. Thus, there exists a convergent subsequence, relabelled as $\{(\bar{x}^k,C^k)\}$, with a limit $(\bar{x}^*,C^*) \in \mathbb{R}^m \times \mathscr{S}_m^+$. Then by the continuity of the functional F in $(\bar{x},C)$, we deduce that $(\bar{x}^*,C^*)$ is a maximiser to $F(\bar{x},C)$,i.e., the existence of a maximiser. The uniqueness follows from the strict joint-concavity of $F(\bar{x},C)$, cf. Theorem 2.3.1. □

Since F is composed of smooth functions, clearly it is smooth. Next we give the gradient formulae, which are useful for developing numerical algorithms below.

Theorem 2.3.3. *The gradients of the functional $F(\bar{x},C)$ with respect to $\bar{x}$ and C are respectively*

given by

$$\frac{\partial F}{\partial \bar{x}} = A^t y - A^t e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)} - C_0^{-1}(\bar{x}-\mu_0),$$
$$\frac{\partial F}{\partial C} = \tfrac{1}{2}[-A^t \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})A - C_0^{-1} + C^{-1}].$$

Proof. Let $d = A\bar{x} + \frac{1}{2}\mathrm{diag}(ACA^t)$. Then by the chain rule

$$\frac{\partial}{\partial \bar{x}_i}(1_n, e^d) = \frac{\partial}{\partial \bar{x}_i}\sum_{j=1}^{n} e^{d_j} = \sum_{j=1}^{n} \frac{\partial e^{d_j}}{\partial d_j}\frac{\partial d_j}{\partial \bar{x}_i} = \sum_{j=1}^{n} e^{d_j}(A)_{ji}.$$

That is, we have $\frac{\partial}{\partial \bar{x}}(1_n, e^d) = A^t e^d$, showing the first formula. Next we derive the gradient with respect to the covariance C. In view of (2.6), it remains to differentiate the term $(1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})$ with respect to C. To this end, let H be a small perturbation to C. By Taylor expansion, and with the diagonal matrix $D = \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})$, we deduce

$$(1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(A(C+H)A^t)}) - (1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)}) = (D, \tfrac{1}{2}\mathrm{diag}(AHA^t)) + \mathscr{O}(\|H\|^2).$$

Since the matrix D is diagonal, by the cyclic property of trace, we have

$$(D, \tfrac{1}{2}\mathrm{diag}(AHA^t)) = (D, \tfrac{1}{2}(AHA^t)) = \tfrac{1}{2}\mathrm{tr}(DAH^tA^t) = \tfrac{1}{2}\mathrm{tr}(A^tDAH^t) = \tfrac{1}{2}(A^tDA, H).$$

Now the definition of the gradient completes the proof. □

An immediate corollary is the following optimality system.

Corollary 2.3.1. *The necessary and sufficient optimality system of problem* (2.7) *is given by*

$$A^t y - A^t e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)} - C_0^{-1}(\bar{x}-\mu_0) = 0,$$
$$C^{-1} - A^t \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})A - C_0^{-1} = 0.$$

Remark 2.3.3. *Challis and Barber [26] showed that for log-concave site posterior potentials, the variational lower bound is jointly concave in $\bar{x}$ and the Cholesky factor L of the covariance C. This assertion holds also for the lower bound $F(\bar{x}, C)$ in* (2.5)*, i.e., joint concavity with respect to $(\bar{x}, L)$.*

Remark 2.3.4. *Corollary 2.3.1 indicates that the covariance C^* of the optimal Gaussian approximation $q^*(\mathbf{x})$ is of the following form:*

$$(C^*)^{-1} = C_0^{-1} + A^t DA,$$

for some diagonal matrix D. Thus it is tempting that one may minimise with respect to D instead of C in order to reduce the complexity of the algorithm, by reducing the number of unknowns from

m^2 to m. However, F is generally not concave with respect to D; see [78] for a one-dimensional counterexample. The loss of concavity might complicate the analysis and computation.

Remark 2.3.5. *In practice, the parameter x in the model* (2.2) *often admits physical constraint. Thus it is natural to impose a box constraint on the mean $\bar{x}$ in problem* (2.7)*, e.g., $c_l \le \bar{x}_i \le c_u$, $i = 1,\dots,m$, for some $c_l < c_u$. This can be easily incorporated into the optimality system in Corollary 2.3.1, and the algorithms below remain valid upon minor changes, e.g., including a pointwise projection operator in the update of $\bar{x}$.*

2.4 Numerical Algorithm and Its Complexity Analysis

Now we develop an efficient numerical algorithm, which is of alternating direction maximisation type, provide an analysis of its complexity, and discuss strategies for complexity reduction.

2.4.1 Numerical Algorithm

In view of the strict concavity of $F(\bar{x},C)$, it suffices to solve the optimality system (cf. Corollary 2.3.1):

$$A^t y - A^t e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)} - C_0^{-1}(\bar{x}-\mu_0) = 0, \tag{2.8}$$

$$C^{-1} - A^t \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})A - C_0^{-1} = 0. \tag{2.9}$$

This consists of a coupled nonlinear system for $(\bar{x},C)$. We shall solve the system by alternatingly maximising $F(\bar{x},C)$ with respect to $\bar{x}$ and C, i.e., coordinate ascent. From the strict concavity in Theorem 2.3.1, we deduce that for a fixed C, (2.8) has a unique solution $\bar{x}$, and similarly, for a fixed $\bar{x}$, (2.9) has a unique solution C. Below, we discuss the efficient numerical solution of (2.8)–(2.9).

2.4.1.1 Newton Method for Updating $\bar{x}$

To solve $\bar{x}$ from (2.8), for a fixed C, we employ a Newton method. Let the nonlinear map $G : \mathbb{R}^m \to \mathbb{R}^m$ be defined by

$$G(\bar{x}) = A^t e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)} + C_0^{-1}(\bar{x}-\mu_0) - A^t y.$$

The Jacobian ∂G of the map G is given by

$$\partial G(\bar{x}) = A^t \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})A + C_0^{-1} \ge C_0^{-1},$$

where the partial ordering $\ge$ is in the sense of symmetric positive definite matrix, i.e., $X \ge Y$ if and only if $X - Y$ is positive semidefinite. That is, the Jacobian $\partial G(\bar{x})$ is uniformly invertible (since the prior covariance C_0^{-1} is invertible). This concurs with the strict concavity of the functional $F(\bar{x},C)$ in $\bar{x}$.

This motivates the use of the Newton method or its variants: for a nonlinear system with uniformly invertible Jacobians, the Newton method converges globally [76]. Specifically, given $\bar{x}^0$, we

iterate

$$\partial G(\bar{x}^k)\delta\bar{x} = -G(\bar{x}^k), \qquad \bar{x}^{k+1} = \bar{x}^k + \delta\bar{x}. \tag{2.10}$$

The main cost of the Newton update (2.10) lies in solving the linear system involving $\partial G(\bar{x}^k)$. Clearly, the Jacobian $\partial G(\bar{x}^k)$ is symmetric and positive definite, and thus the (preconditioned) conjugate gradient method is a natural choice for solving the linear system. One may use C_0^{-1} (or the diagonal part of the Jacobian $\partial G(\bar{x})$) as a preconditioner. It is worth noting that inverting the Jacobian $\partial G(\bar{x})$ is identical with one fixed point update of the covariance C below. In the presence of *a priori* structural information, this can be carried out efficiently even for very large-scale problems; see Section 2.4.2 below for further details. By the fast local convergence of the Newton method, a few iterations suffice the desired accuracy, which is fully confirmed by our numerical experiments.

2.4.1.2 Fixed-point Method for Updating C

Next we turn to the solution of (2.9) for updating C, with $\bar{x}$ fixed. There are several different strategies, and we shall describe two of them below. The first option is to employ a Newton method. Let the nonlinear map $S : \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times m}$ be defined by

$$S(C) = C^{-1} - C_0^{-1} - A^t \mathrm{diag}(e^{A\bar{x}+\mathrm{diag}(ACA^t)})A.$$

The Jacobian ∂S of the map S is given by

$$\partial S(C)[H] = -C^{-1}HC^{-1} - A^t \mathrm{diag}(e^{A\bar{x}+\mathrm{diag}(ACA^t)})\mathrm{diag}(AHA^t)A. \tag{2.11}$$

It can be verified that the map $\partial S(C)$ is symmetric with a uniformly bounded inverse (see the proof of Theorem A.2.1 in the appendix for details). However, its explicit form seems not available due to the presence of the operator diag. Nonetheless, one can apply a (preconditioned) conjugate gradient method for updating C. The Newton iteration is guaranteed to converge globally.

The second option is to use a fixed-point iteration. This choice is very attractive since it avoids solving huge linear systems. Specifically, given an initial guess C^0, we iterate by

$$D^k = \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(AC^kA^t)}), \qquad C^{k+1} = (C_0^{-1} + A^t D^k A)^{-1}. \tag{2.12}$$

Conceptually, it has the flavour of a classical fixed point scheme for solving algebraic Riccati equations in Kalman filtering [9], and it has also been used in a slightly different context of variational inference with Gaussian processes [78]. Numerically, each inner iteration of (2.12) involves computing the vector $\mathrm{diag}(AC^kA^t)$ (which should be regarded as computing $A_iC^kA_i^t$, $i = 1,\ldots,m$, instead of forming the full matrix AC^kA^t) and a matrix inversion.

Remark 2.4.1. *From the iteration scheme* (2.12)*, one can see that the variable C^k remains symmetric positive definite during iterations. Note that the Equation* (2.11) *is of a function form, its tensor form*

is of size $m^2 \times m^2$ and thus leads to expensive inversion, which motivates the fixed point scheme for C updating.

Next we briefly discuss the convergence of (2.12). Clearly, for all iterates C^k, we have $C^k \leq C_0$. We claim $\mu_{\max}(C^k) \leq \mu_{\max}(C_0)$. To see this, let $v \in \mathbb{R}^m$ be a unit eigenvector corresponding to the largest eigenvalue $\mu_{\max}(C^k)$, i.e., $v^t C^k v = \mu_{\max}(C^k)$. Then by the minmax principle

$$\mu_{\max}(C^k) = v^t C^k v \leq v^t C_0 v \leq \sup_{v \in \mathbb{S}^m} v^t C_0 v = \mu_{\max}(C_0).$$

Thus, the sequence $\{C^k\}_{k=1}^{\infty}$ generated by the iteration (2.12) is uniformly bounded in the spectral norm (and thus any norm due to the norm equivalence in a finite-dimensional space). Hence, there exists a convergent subsequence, also relabelled as $\{C^k\}$, such that $C^k \to C^*$, for some C^*. In practice, the iterates converge fairly steadily to the unique solution to (2.9), which however remains to be established. In Appendix A.1, we show a certain "monotone" type convergence of (2.12) for the initial guess $C^0 = C_0$.

2.4.1.3 Variational Gaussian Approximation Algorithm

With the preceding two inner solvers, we are ready to state the complete procedure in Algorithm 1. One natural stopping criterion at Step 7 is to monitor ELBO. However, computing ELBO can be expensive and cheap alternatives, e.g., relative change of the mean $\bar{x}$, might be considered. Note that Step 3 of Algorithm 1, i.e., randomised singular value decomposition (rSVD), has to be carried out only once, and it constitutes a preprocessing step. Its crucial role will be discussed in Section 2.4.2 below.

With exact inner updates $(\bar{x}^k, C^k)$, by the alternating maximising property, the sequence $\{F(\bar{x}^k, C^k)\}$ is guaranteed to be monotonically increasing, i.e.,

$$F(\bar{x}^0, C^0) \leq F(\bar{x}^1, C^0) \leq F(\bar{x}^1, C^1) \leq \ldots \leq F(\bar{x}^k, C^k) \leq \ldots,$$

with the inequality being strict until convergence is reached. Further, $F(\bar{x}^k, C^k) \leq \ln Z(y)$. Thus, $\{F(\bar{x}^k, C^k)\}$ converges. Further, by [18, Prop. 2.7.1], the coordinate ascent method converges if the maximisation with respect to each coordinate is uniquely attained. Clearly, Algorithm 1 is a coordinate ascent method for $F(\bar{x}, C)$, and $F(\bar{x}, C)$ satisfies the unique solvability condition. Thus the sequence $\{(\bar{x}^k, C^k)\}$ generated by Algorithm 1 converges to the unique maximiser of $F(\bar{x}, C)$.

2.4.2 Complexity Analysis and Reduction

Now we analyse the computational complexity of Algorithm 1, and describe strategies for complexity reduction, in order to arrive at a scalable implementation. When evaluating the functional $F(\bar{x}, C)$ and its gradient, the constant terms can be precomputed. Thus, it suffices to analyse the terms that will be updated. Standard linear algebra [54] gives the following operational complexity.

Algorithm 1 Variational Gaussian Approximation Algorithm

1: Input: (A, y), specify the prior (μ_0, C_0), and the maximum number K of iterations
2: Initialise $\bar{x} = \bar{x}^1$ and $C = C^1$;
3: SVD: $(U, \Sigma, V) = \text{rSVD}(A)$;
4: **for** $k = 1, 2, \ldots, K$ **do**
5: Update the mean $\bar{x}^{k+1}$ by (2.10);
6: Update the covariance C^{k+1} by (2.12);
7: Check the stopping criterion.
8: **end for**
9: Output: $(\bar{x}, C)$

- The complexity of evaluating the objective functional $F(\bar{x}, C)$ is $\mathcal{O}(m^2 n + m^3)$:
  - the inner product $-(1_n, e^{A\bar{x} + \frac{1}{2}\text{diag}(ACA^t)}) \sim \mathcal{O}(m^2 n)$
  - the matrix determinant $\ln|C| \sim \mathcal{O}(m^3)$
- The complexity of evaluating the gradient $\frac{\partial F}{\partial \bar{x}}$ is $\mathcal{O}(m^2 n)$:
  - the matrix-vector product $A^t e^{A\bar{x} + \frac{1}{2}\text{diag}(ACA^t)} \sim \mathcal{O}(m^2 n)$
- The complexity of evaluating the gradient $\frac{\partial F}{\partial C}$ is $\mathcal{O}(m^2 n + m^3)$:
  - the matrix product $A^t \text{diag}(e^{A\bar{x} + \frac{1}{2}\text{diag}(ACA^t)}) A \sim \mathcal{O}(m^2 n)$
  - the matrix inversion $C^{-1} \sim \mathcal{O}(m^3)$.

In summary, evaluating ELBO $F(\bar{x}, C)$ and its gradients each involves $\mathcal{O}(nm^2 + m^3)$ complexity, which is infeasible for large-scale problems. The most expensive piece lies in matrix products/inversion, e.g., $(1_n, e^{A\bar{x} + \frac{1}{2}\text{diag}(ACA^t)})$, $A^t e^{A\bar{x} + \frac{1}{2}\text{diag}(ACA^t)}$ and $A^t \text{diag}(e^{A\bar{x} + \frac{1}{2}\text{diag}(ACA^t)}) A$. The log-determinant $\ln|C|$ can be approximated accurately with $\mathcal{O}(m^2)$ operations by a stochastic algorithm [144]. In many practical inverse problems, there do exist structures: (i) A is low rank, and (ii) C is sparse, which can be leveraged to reduce the per-iteration cost.

First, for many inverse problems, the matrix A is ill-conditioned, and the singular values decay to zero. Thus, A naturally has a low-rank structure. The effective rank r is determined by the decay rate of the singular values. In this work, we assume a known rank r. The rSVD is a powerful technique for obtaining low-rank approximations [59]. For a rank r matrix, the rSVD can yield an accurate approximation with $\mathcal{O}(mn \ln r + (m+n)r^2)$ operations [59, pp. 225]. We denote the rSVD approximation by $A \approx U\Sigma V^t$, where the matrices $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{m \times r}$ are column orthonormal, and $\Sigma \in \mathbb{R}^{r \times r}$ is diagonal with its entries ordered nonincreasingly.

Second, the covariance C is approximately sparse, and each row/column has at most s nonzero entries. This reflects the fact that only (physically) neighbouring elements are highly correlated, and there is no long range correlation. This choice will be implemented in the numerical experiments for 2D image deblurring. Naturally, one can also consider a more flexible option by adaptively selecting the sparsity pattern. This can be achieved by penalising of the off-diagonal entries of C

by the ℓ^1-norm, which allows automatically detecting significant correlation [113]. Other structures, e.g., low-rank plus sparsity, offer potential alternatives. Note that the choice of covariance structures does depend on specific applications and our choice is more suitable for problems such as image deblurring.

Under these structural assumptions, the complexity of computing the terms $(1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})$, $A^t e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)}$ and $A^t\mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})A$ can be reduced to $\mathscr{O}(smn)$. Thus, the complexity of calculating F and $\frac{\partial F}{\partial \bar{x}}$ is reduced to $\mathscr{O}(smn+m^2)$. For the matrix inversion in (2.12), we exploit the low-rank structure of A. Upon recalling the low-rank approximation of A and the Sherman-Morrison-Woodbury formula [54, pp. 65], i.e.,

$$(\tilde{A}+\tilde{U}\tilde{V})^{-1} = \tilde{A}^{-1} - \tilde{A}^{-1}\tilde{U}(I+\tilde{V}\tilde{A}^{-1}\tilde{U})^{-1}\tilde{V}\tilde{A}^{-1},$$

we deduce (with $D = \mathrm{diag}(e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)})$)

$$C = C_0 - C_0 V\Sigma U^t DU\Sigma(I + V^t C_0 V\Sigma U^t DU\Sigma)^{-1}V^t C_0. \tag{2.13}$$

Note that the inversion step only involves a matrix in $\mathbb{R}^{r\times r}$, and can be carried out efficiently. The sparsity structure on C can be enforced by computing only the respective entries. Then the update formula (2.13) can be achieved in $\mathscr{O}(smn+r^2n+r^2m)$ operations. In comparison with the $\mathscr{O}(m^3+nm^2)$ complexity of the direct implementation, this represents a substantial complexity reduction.

2.5 Hyperparameter Choice with Empirical Bayes

When encoding prior knowledge about the unknown x into the prior $p(x)$, it is often necessary to tune its strength, a scalar parameter commonly known as hyperparameter. It plays the role of the regularisation parameter in variational regularisation [67, Chapter 7], where its proper choice is notoriously challenging. In the Gaussian prior $p(x)$, $C_0 = \alpha^{-1}\bar{C}_0$, where $\bar{C}_0$ describes the interaction structure and the scalar α determines the strength of the interaction which has to be specified.

In the Bayesian paradigm, one principled approach to handle hyperparameters is Empirical Bayes [118], by estimating the hyperparameter in the prior distribution from data. Another popular principled approach for hyperparameter selection in the Bayesian community is hierarchical Bayesian model [118] which regards the hyperparameter as a random variable and stipulates a hyperprior distribution on the hyperparameter. While the connection between these two approaches has been studied in the literature [107], in this work, we recover the empirical Bayes for our studied model from similar starting point as hierarchical model. Specifically, we write the Gaussian prior $p(x|\alpha) = \mathscr{N}(x|0,\alpha^{-1}\bar{C}_0)$, and employ a Gamma distribution $p(\alpha|a,b) = \mathrm{Gamma}(\alpha|a,b)$ on α, where (a,b) are the parameters. The Gamma distribution is the conjugate prior for α, and it is analytically and computationally convenient. In practice, one may take (a,b) close to $(1,0)$ to mimic

a noninformative prior. Then appealing to Bayes' formula again, one obtains a posterior distribution (jointly over (x,α)). Conceptually, with the VGA, this construction determines the optimal parameter by maximising ELBO as a function of α, i.e., model selection within a parametric family. Thus it can be viewed as a direct application of ELBO in model selection.

One may explore the resulting joint posterior distribution in several ways [67, Chapter 7]. In this work, we employ an EM type method to maximise the following (joint) lower bound

$$\begin{aligned} F(\bar{x},C,\alpha) &= \int q(x)\ln\frac{p(x,y|\alpha)p(\alpha|a,b)}{q(x)}\mathrm{d}x \\ &= \int q(x)\ln\frac{p(x,y|\alpha)}{q(x)}\mathrm{d}x + \int q(x)\ln p(\alpha|a,b)\mathrm{d}x \\ &= F_\alpha(\bar{x},C) + (a-1)\ln\alpha - \alpha b + \ln\frac{b^a}{\Gamma(a)}, \end{aligned}$$

where the subscript α indicates the dependence of ELBO on α. Then, using (2.5) and substituting C_0 with $\alpha^{-1}\bar{C}_0$, we have

$$\begin{aligned} F(\bar{x},C,\alpha) = (y,A\bar{x}) - (1_n, e^{A\bar{x}+\frac{1}{2}\mathrm{diag}(ACA^t)}) - \tfrac{\alpha}{2}(\bar{x}-\mu_0)^t\bar{C}_0^{-1}(\bar{x}-\mu_0) - \tfrac{\alpha}{2}\mathrm{tr}(\bar{C}_0^{-1}C) \\ + \tfrac{1}{2}\ln|C| + \tfrac{m}{2}\ln\alpha - \tfrac{1}{2}\ln|\bar{C}_0| + (a-1)\ln\alpha - \alpha b + \tfrac{m}{2} - (1_n,\ln(y!)) + \ln\frac{b^a}{\Gamma(a)}. \end{aligned} \tag{2.14}$$

This functional extends ELBO $F(\bar{x},C)$ to estimate the hyperparameter α simultaneously with $(\bar{x},C)$ in a way analogous to augmented Tikhonov regularisation [71].

To maximize $F(\bar{x},C,\alpha)$, we employ an EM algorithm [19, Chapter 9.3]. In the E-step, we fix α, and maximise $F(\bar{x},C,\alpha)$ for a new Gaussian approximation $\mathcal{N}(x|\bar{x},C)$ by Algorithm 1, with the unique maximiser denoted by $(\bar{x}_\alpha,C_\alpha)$. Then in the M-step, we fix $(\bar{x},C)$ and update α by

$$\alpha = \frac{m+2(a-1)}{(\bar{x}_\alpha-\mu_0)^t\bar{C}_0^{-1}(\bar{x}_\alpha-\mu_0)+\mathrm{tr}(\bar{C}_0^{-1}C_\alpha)+2b}. \tag{2.15}$$

This follows from the condition $\frac{\partial F}{\partial\alpha}=0$. These discussions lead to the procedure in Algorithm 2. A natural stopping criterion at line 5 is the change of α. Below we analyse the convergence of Algorithm 2.

Remark 2.5.1. *The first two terms in the denominator of the iteration* (2.15) *is given by*

$$\alpha(\bar{x}_\alpha-\mu_0)^t\bar{C}_0^{-1}(\bar{x}_\alpha-\mu_0)+\alpha\mathrm{tr}(\bar{C}_0^{-1}C_\alpha) = \mathbb{E}_{q(x)}[\|x-\mu_0\|^2_{C_0^{-1}}],$$

i.e., the expectation of the negative logarithm of the Gaussian prior $p(x)$ with respect to the Gaussian posterior approximation $q(x)$. Formally, the fixed point iteration (2.15) *can be viewed as an extension of that for a balancing principle for Tikhonov regularisation in [71, 68] to a probabilistic context.*

Algorithm 2 Variational Gaussian approximation with Empirical Bayes

1: Input (A, y), and initialise α^1
2: **for** $k = 1, 2, \ldots$ **do**
3: E-step: Update $(\bar{x}^k, C^k)$ by Algorithm 1:

$$(\bar{x}^k, C^k) = \arg \max_{(\bar{x}, C) \in \mathbb{R}^m \times \mathscr{S}_m^+} F_{\alpha^k}(\bar{x}, C);$$

4: M-step: Update α by (2.15).
5: Check the stopping criterion;
6: **end for**
7: Output: $(\bar{x}, C)$

In order to analyse the convergence of Algorithm 2, we write the functional $F_\alpha(\bar{x}, C)$ as

$$F_\alpha(\bar{x}, C) = \phi(\bar{x}, C) + \alpha\psi(\bar{x}, C),$$

where

$$\begin{aligned}
\phi(\bar{x}, C) &= (y, A\bar{x}) - (1_n, e^{A\bar{x} + \frac{1}{2}\mathrm{diag}(ACA^t)}) + \tfrac{1}{2}\ln|C| - \tfrac{1}{2}\ln|\bar{C}_0| + -(1_n, \ln(y!)), \\
\psi(\bar{x}, C) &= -\tfrac{1}{2}(\bar{x} - \mu_0)^t \bar{C}_0^{-1}(\bar{x} - \mu_0) - \tfrac{1}{2}\mathrm{tr}(\bar{C}_0^{-1} C) \leq 0.
\end{aligned}$$

Thus the functional $F_\alpha(\bar{x}, C)$ resembles classical Tikhonov regularisation. By Theorem 2.3.2, for any $\alpha > 0$, there exists a unique maximiser $(\bar{x}_\alpha, C_\alpha)$ to F_α, and the value function $\psi(\bar{x}_\alpha, C_\alpha)$ is continuous in α, cf. Lemma 2.5.2 below. In Appendix A.2, we show that the maximiser $(\bar{x}_\alpha, C_\alpha)$ is actually differentiable in α.

Lemma 2.5.1. *For any $\alpha > 0$, the maximiser $(\bar{x}_\alpha, C_\alpha)$ is bounded, with the bound depending only on α.*

Proof. Taking inner product between (2.8) and $\bar{x}_\alpha$, we deduce

$$(C_0^{-1}\bar{x}_\alpha, \bar{x}_\alpha) + (e^{A\bar{x}_\alpha + \mathrm{diag}(ACA^t)}, A\bar{x}_\alpha) = (A^t y, \bar{x}_\alpha).$$

It can be verified directly that the function $f(t) = te^t$ is bounded from below by $-e^{-1}$ for $t \in \mathbb{R}$. Meanwhile, by (2.9), $C \leq C_0$, and thus

$$(e^{A\bar{x}_\alpha + \mathrm{diag}(ACA^t)}, A\bar{x}_\alpha) \geq -e^{-1}\sum_i e^{\mathrm{diag}(ACA^t)_i} \geq -e^{-1}\sum_i e^{\mathrm{diag}(AC_0A^t)_i} = -ce^{-1}.$$

This and the Cauchy-Schwarz inequality give $\|\bar{x}_\alpha\| \leq c\alpha^{-1}$, with c depending only on y. Next, by (2.9), we have

$$0 \leq e^{(A\bar{x})_i + \mathrm{diag}(ACA^t)_i} \leq e^{(A\bar{x})_i + \mathrm{diag}(AC_0A^t)_i} \leq c,$$

and consequently appealing to (2.9) again yields $(C_0^{-1}+cA^tA)^{-1} \le C \le C_0$, completing the proof. □

Lemma 2.5.2. *The functional value $\psi(\bar{x}_\alpha, C_\alpha)$ is continuous at any $\alpha > 0$.*

Proof. Let $\{\alpha^k\} \subset \mathbb{R}^+$ be a sequence convergent to α. By Theorem 2.3.2, for each α^k, there exists a unique maximiser $(\bar{x}^k, C^k)$ to $F_{\alpha^k}(\bar{x}, C)$. By Lemma 2.5.1, the sequence $\{(\bar{x}^k, C^k)\}$ is uniformly bounded, and there exists a convergent subsequence, relabelled as $\{(\bar{x}^k, C^k)\}$, with a limit $(\bar{x}^*, C^*)$. By the continuity of the functionals $\phi(\bar{x}, C)$ and $\psi(\bar{x}, C)$, we have for any $(\bar{x}, C) \in \mathbb{R}^m \times \mathscr{S}_m^+$

$$\begin{aligned} F_\alpha(\bar{x}^*, C^*) &= \lim_{k\to\infty} (\phi(\bar{x}^k, C^k) + \alpha_k \psi(\bar{x}^k, C^k)) \ge \lim_{k\to\infty} (\phi(\bar{x}, C) + \alpha_k \psi(\bar{x}, C)) \\ &= \phi(\bar{x}, C) + \alpha \psi(\bar{x}, C) = F_\alpha(\bar{x}, C). \end{aligned}$$

That is, $(\bar{x}^*, C^*)$ is a maximiser of $F_\alpha(\bar{x}, C)$. The uniqueness of the maximiser to $F_\alpha(\bar{x}, C)$ and a standard subsequence argument imply that the whole sequence converges. The desired continuity now follows by the continuity of $\psi(\bar{x}, C)$ in $(\bar{x}, C)$. □

Next we give an important monotonicity relation for $\psi(\bar{x}_\alpha, C_\alpha)$ in α, in a manner similar to classical Tikhonov regularisation [68]. In Appendix A.2, we show that it is actually strictly monotone.

Lemma 2.5.3. *The functional $\psi(\bar{x}_\alpha, C_\alpha)$ is monotonically increasing in α.*

Proof. This result follows by a standard comparison principle. For any α_1, α_2, by the maximising property of $(C_{\alpha_1}, \bar{x}_{\alpha_1})$ and $(C_{\alpha_2}, \bar{x}_{\alpha_2})$, we have

$$F_{\alpha_1}(\bar{x}_{\alpha_1}, C_{\alpha_1}) \ge F_{\alpha_1}(\bar{x}_{\alpha_2}, C_{\alpha_2}) \quad \text{and} \quad F_{\alpha_2}(\bar{x}_{\alpha_2}, C_{\alpha_2}) \ge F_{\alpha_2}(\bar{x}_{\alpha_1}, C_{\alpha_1}).$$

Summing up these two inequalities and collecting terms yield

$$(\alpha_1 - \alpha_2)[\psi(\bar{x}_{\alpha_1}, C_{\alpha_1}) - \psi(\bar{x}_{\alpha_2}, C_{\alpha_2})] \ge 0.$$

Then the desired monotonicity relation follows. □

Theorem 2.5.1. *For any initial guess $\alpha^1 > 0$, the sequence $\{\alpha^k\}$ generated by Algorithm 2 is monotonically convergent to some $\alpha^* \ge 0$, and if the limit $\alpha^* > 0$, then it satisfies the fixed point equation* (2.15).

Proof. By the fixed point iteration (2.15), we have (with $c = \frac{m}{2} + a - 1$)

$$\alpha^{k+1} - \alpha^k = \frac{c}{-\psi(\bar{x}_{\alpha^k}, C_{\alpha^k}) + b} - \frac{c}{-\psi(\bar{x}_{\alpha^{k-1}}, C_{\alpha^{k-1}}) + b}$$

$$= \frac{c[\psi(\bar{x}_{\alpha^k}, C_{\alpha^k}) - \psi(\bar{x}_{\alpha^{k-1}}, C_{\alpha^{k-1}})]}{(-\psi(\bar{x}_{\alpha^k}, C_{\alpha^k}) + b)(-\psi(\bar{x}_{\alpha^{k-1}}, C_{\alpha^{k-1}}) + b)}.$$

Since $\psi \leq 0$, the denominator is positive. By Lemma 2.5.3, $\alpha^{k+1} - \alpha^k$ and $\alpha^k - \alpha^{k-1}$ have the same sign, and thus $\{\alpha^k\}$ is monotone. Further, for all α^k, we have $0 \leq \alpha^k \leq \frac{m+2(a-1)}{2b}$, i.e., $\{\alpha^k\}$ is uniformly bounded. Thus $\{\alpha^k\}$ is convergent. By Lemma 2.5.2, $\psi(\bar{x}_\alpha, C_\alpha)$ is continuous in α for $\alpha > 0$, and α^* satisfies (2.15). □

Remark 2.5.2. *The proof of Theorem 2.5.1 provides a constructive approach to the existence of a solution to* (2.15)*. The uniqueness of the solution α^* to* (2.15) *is generally not ensured. However, in practice, it seems to have only two fixed points: one is in the neighbourhood of $+\infty$, which is uninteresting, and the other is the desired one.*

2.6 Numerical Experiments and Discussions

Now we present numerical results to examine algorithmic features (Sections 2.6.1–2.6.4, with the example `phillips`) and to illustrate the VGA (Section 2.6.5). All one-dimensional examples are taken from public domain `MATLAB` package `Regutools`[1], and the discrete problems are of size 100×100. We refer the prior with a zero mean $\mu_0 = 0$ and the covariance $\alpha^{-1} I_m$ and $\alpha^{-1} L_1^{-1} L_1^{-t}$ (with L_1 being the 1D first-order forward difference matrix) to as the L^2- and H^1-prior, respectively, and let $\bar{C}_0 = I_m$, and $\bar{C}_1 = L_1^{-1} L_1^{-t}$. Unless otherwise specified, the parameter α is determined in a trial-and-error manner, and in Algorithm 1, the Newton update $\delta\bar{x}$ in (2.10) is computed by the `MATLAB` built-in function `pcg` with a default tolerance, the prior covariance C_0^{-1} as the preconditioner and a maximum 10 PCG iterations.

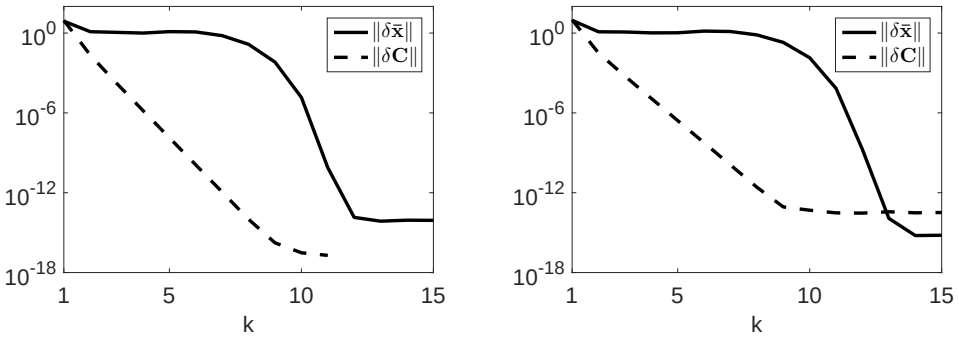
Remark 2.6.1. *Note that since the difference operator L_1 has non-zero null space, shifting any point with any element in the null space will not change the density. As a result, the integration of the prior on the whole space will diverge. In other words, the H^1-seminorm prior is not a proper prior. However, this problem will not occur in the joint distribution, since the log density of Poisson likelihood with exponential inverse link function does not vanish this kernel. Further, the evidence $Z(y)$ will not diverge given the improperness of the prior. Since the ELBO functional only relies on the well-definedness of the joint distribution, the VGA framework is applicable.*

2.6.1 Convergence Behaviour of Inner and Outer Iterations of Algorithm 1

First, we examine the convergence behaviour of inner iterations for updating $\bar{x}$ and C, i.e., (2.10) and (2.12), for the example `phillips` with the L^2-prior $C_0 = 1.0 \times 10^{-1} \bar{C}_0$ and H^1-prior $C_0 = 2.5 \times 10^{-3} \bar{C}_1$. To study the convergence, we fix C at $C^1 = I$ for $\bar{x}$ and present the ℓ^2-norm of the update $\delta\bar{x}$ (initialised with $\bar{x}^0 = 0$), and similarly fix $\bar{x}$ at the converged iterate $\bar{x}^1$ for C and present the spectral norm of the change δC. For both (2.10) and (2.12), these initial guesses are quite far

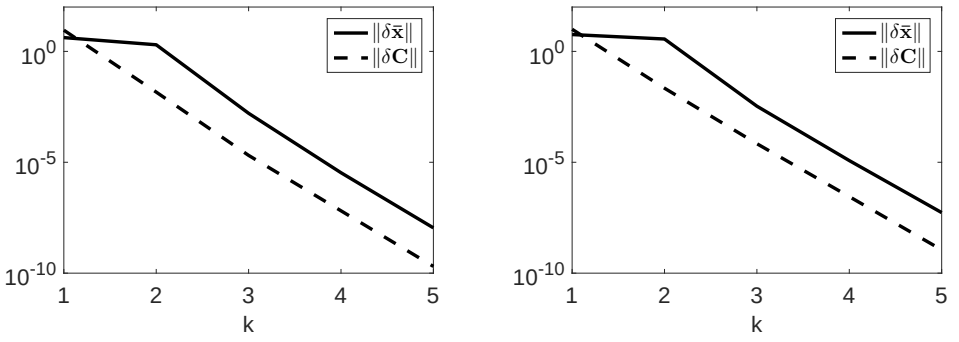
[1] `http://www.imm.dtu.dk/~pcha/Regutools/`, last accessed on April 15, 2017

away from the solutions, and thus the choice allows showing their global convergence behaviour. The convergence is fairly rapid and steady for both inner iterations, cf. Fig. 2.1. For example, for a tolerance 10^{-5}, the Newton method (2.10) converges after about 10 iterations, and the fixed point method (2.12) converges after 4 iterations, respectively. The global as well as local superlinear convergence of the Newton method (2.10) are clearly observed, confirming the discussions in Section 2.4. The convergence behaviour of the inner iterations is similar for both priors. In practice, it is unnecessary to solve the inner iterates to a very high accuracy, and it suffices to apply a few inner updates within each outer iteration. Since the iteration (2.12) often converges faster than (2.8), we take five Newton updates and one fixed point update per outer iteration for the numerical experiments below.



(a) L^2-prior (b) H^1-prior

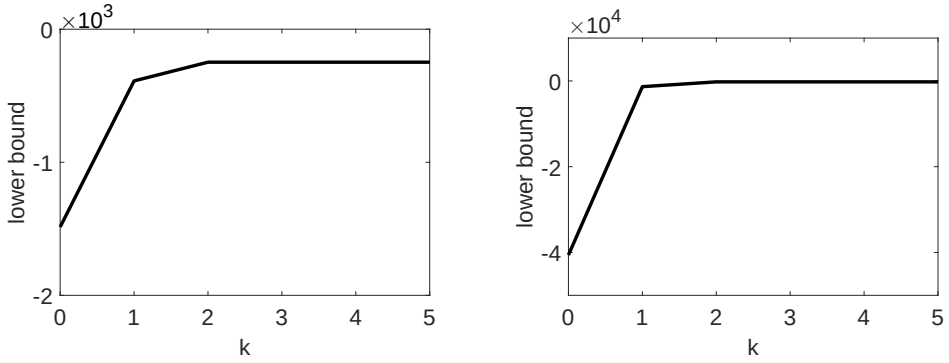
Figure 2.1: The convergence of the inner iterations of Algorithm 1 for `phillips`.



(a) L^2 prior (b) H^1-prior

Figure 2.2: The convergence of outer iterations of Algorithm 1 for `phillips`.

To examine the convergence of outer iterations, we show the errors of the mean $\bar{x}$ and covariance C and the lower bound $F(\bar{x},C)$ in Figs. 2.2 and 2.3, respectively. Algorithm 1 is terminated when the change of the lower bound falls below 10^{-10}. For the L^2-prior, Algorithm 1 converges after 5 iterations and the last increments $\delta\bar{x}$ and δC are of order 10^{-8} and 10^{-9}, respectively. This observation holds also for the H^1-prior, cf. Figs. 2.2(b) and 2.3(b). Thus, both inner and outer



(a) L^2-prior (b) H^1-prior

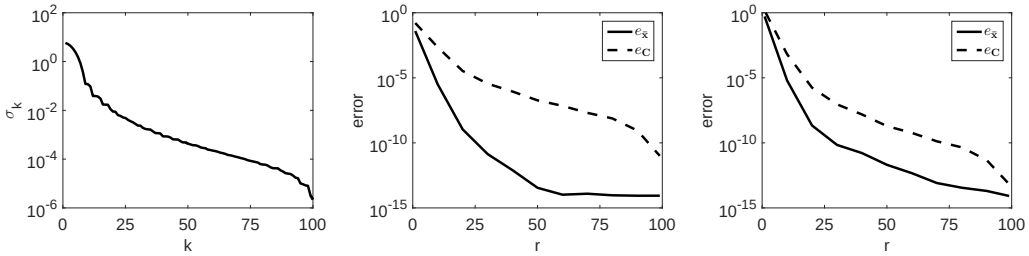
Figure 2.3: The convergence of the lower bound $F(\bar{x},C)$ for `phillips`.

iterations converge rapidly and steadily, and Algorithm 1 is very efficient.

2.6.2 Low-rank Approximation of A and Sparsity of C

The discussions in Section 2.4.2 show that the structure on A and C can be leveraged to reduce the complexity of Algorithm 1. Now we evaluate their influence on the accuracy of the VGA.

First, we examine the influence of low-rank approximation to A. Since the kernel function of the example `phillips` is smooth, the inverse problem is mildly ill-posed and the singular values σ_k decay algebraically, cf. Fig. 2.4(a). A low-rank matrix A_r of rank $r \approx 10$ can already approximate A well. To study its influence on the VGA, we denote by $(\bar{x}_r, C_r)$ and $(\bar{x}^*, C^*)$ the VGA for A_r and A, respectively. The errors $e_{\bar{x}} = \|\bar{x}_r - \bar{x}^*\|$ and $e_C = \|C_r - C^*\|$ for different ranks r are shown in Figs. 2.4 (b) and (c) for the L^2- and H^1-prior, respectively. Too small a rank r of the approximation A_r can lead to pronounced errors in both the mean $\bar{x}$ and the covariance C, whereas for a rank of $r = 10$, the errors already fall below one percent. Interestingly, the decay of the error $e_{\bar{x}}$ is much faster than that of the singular values σ_k, and the error e_C decays slower than $e_{\bar{x}}$. The fast decay of the errors $e_{\bar{x}}$ and e_C indicates the robustness of the VGA, which justifies using low-rank approximations in Algorithm 1.



(a) singular values σ_k (b) L^2-prior (c) H^1-prior

Figure 2.4: (a) singular values and (b)–(c): the errors of the mean and covariance for `phillips`.

Next we examine the influence of the sparsity assumption on the covariance C, which is used to reduce the complexity of Algorithm 1. Due to the coupling between $\bar{x}$ and C, cf. (2.8)–(2.9), the

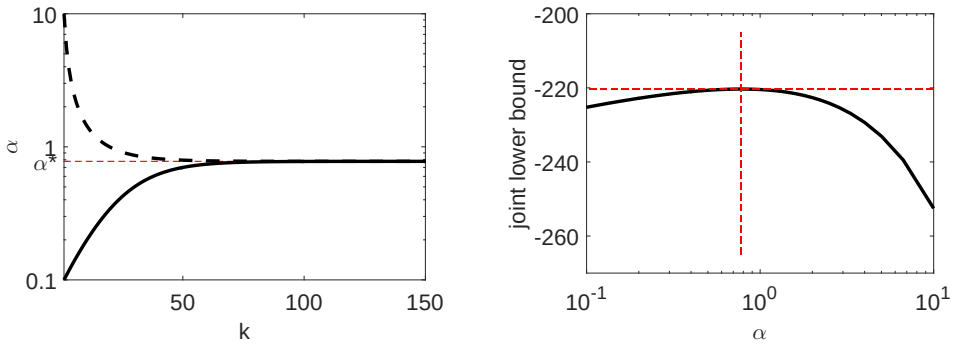
sparsity assumption on C affects the accuracy of both $\bar{x}$ and C. To illustrate this, we take different sparsity levels s on C in Algorithm 1, i.e., at most s nonzero entries around the diagonal of C. Surprisingly, a diagonal C already gives an acceptable approximation measured by the errors $e_{\bar{x}} = \|\bar{x}_s - \bar{x}^*\|_2$ and $e_C = \|C_s - C^*\|_2$, where $(\bar{x}_s, C_s)$ is the VGA with a sparsity level s. The errors $e_{\bar{x}}$ and e_C decrease with the sparsity level s, cf. Table 2.1. Thus the sparsity assumption on C can reduce significantly the complexity while retaining the accuracy.

Table 2.1: The errors $e_{\bar{x}}$ and e_C v.s. the sparsity level s of C for `phillips`.

prior	L^2 prior		H^1 prior	
s	$e_{\bar{x}}$	e_C	$e_{\bar{x}}$	e_C
1	6.38e-2	9.20e-2	1.92e-2	7.06e-2
3	5.62e-2	8.10e-2	1.27e-2	5.42e-2
5	4.88e-2	7.02e-2	1.00e-2	4.29e-2

2.6.3 Parameter Choice

Now we examine the convergence of Algorithm 2 for choosing the parameter α in the prior $p(x)$. By Theorem 2.5.1, the sequence $\{\alpha^k\}$ generated by Algorithm 2 is monotone. We illustrate this by two initial guesses, i.e., $\alpha^1 = 0.1$ and $\alpha^1 = 10$. Both sequences of iterates generated by Algorithm 2 converge monotonically to the limit $\alpha^* = 0.7778$, and the convergence of Algorithm 2 is fairly steady, cf. Fig. 2.5(a). Further, Algorithm 2 indeed maximises the joint lower bound (2.14) with its maximum attained at $\alpha^* = 0.7778$, cf. Fig. 2.5(b). Though not shown, the lower bound $F_\alpha(\bar{x}, C|\alpha)$ is also increasing during the iteration. Thus, the empirical Bayesian approach is indeed performing model selection by maximising ELBO.



(a) convergence of α (b) joint lower bound

Figure 2.5: (a)The convergence of Algorithm 2 initialised with 0.1 and 10, both convergent to $\alpha^* = 0.7778$ (b) the joint lower bound versus α, for `phillips` with L^2-prior.

To illustrate the quality of the automatically chosen parameter α, we take six realisations of the Poisson data y and compare the mean $\bar{x}$ of the VGA with the optimal regularised solutions, where α is tuned so that the error is smallest (and thus it is infeasible in practice). The means $\bar{x}$ by Algorithm 2 are comparable with the optimal ones, cf. Fig. 2.6, and thus the empirical Bayesian approach can

yield reasonable approximations. The parameter α by the empirical Bayesian approach is slightly smaller than the optimal one, cf. Table 2.2, and hence the corresponding reconstruction tends to be slightly more oscillatory than the optimal one. The value of the parameter α by the empirical Bayesian approach is relatively independent of the realisation, whose precise mechanism is to be ascertained.

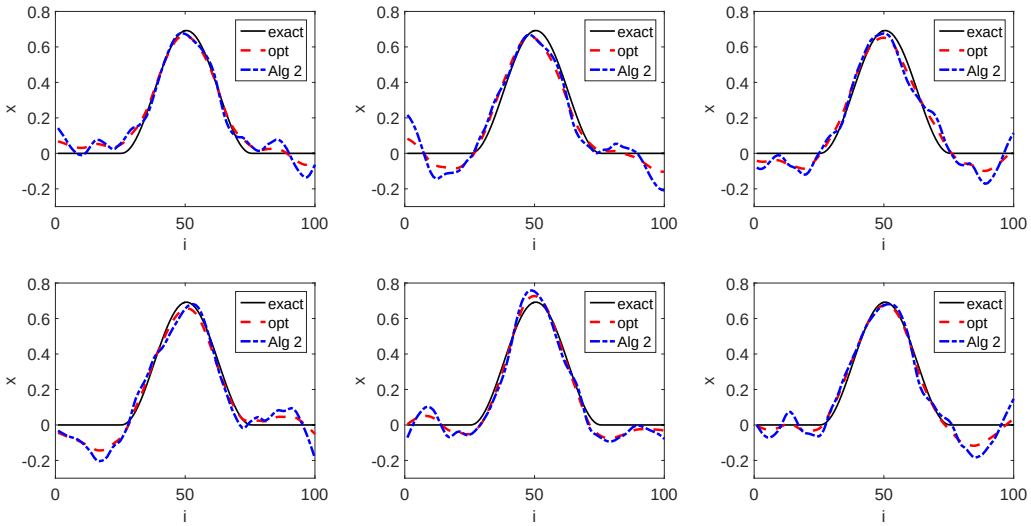


Figure 2.6: The mean $\bar{x}$ of the Gaussian approximation by Algorithm 2 (Alg2) and the "optimal" solution (opt) for 6 realisations of Poisson data for `phillips` with the L^2-prior.

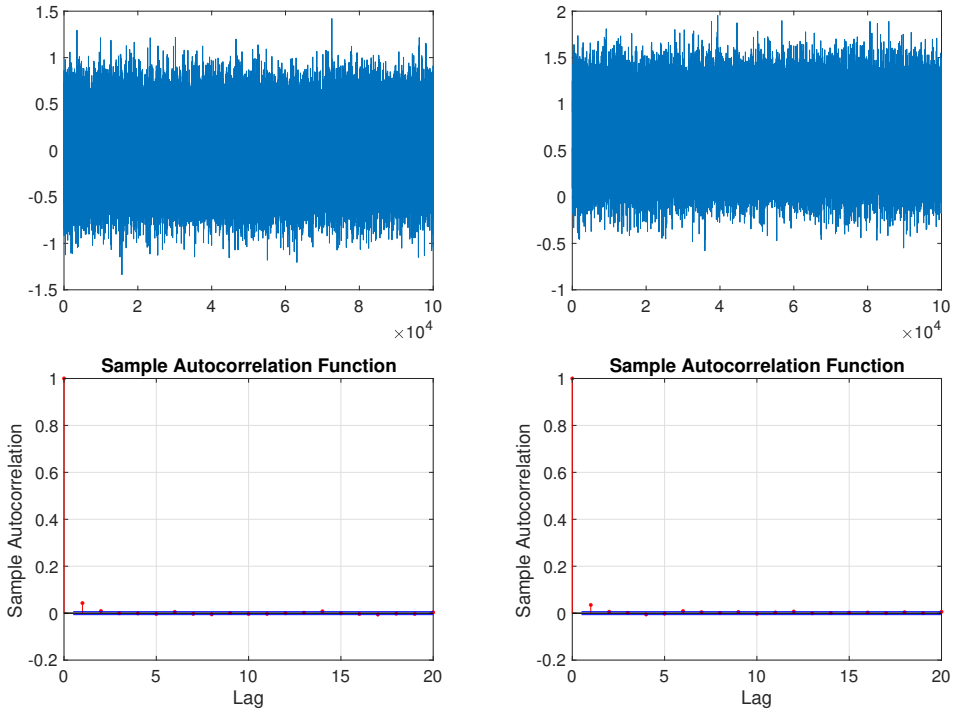
Table 2.2: The values of the hyperparameter α for the results in Fig. 2.6.

case	1	2	3	4	5	6
opt	2.64	3.35	2.59	1.35	9.31	4.04
Alg 2	0.78	0.76	0.76	0.77	0.73	0.74

2.6.4 VGA versus MCMC

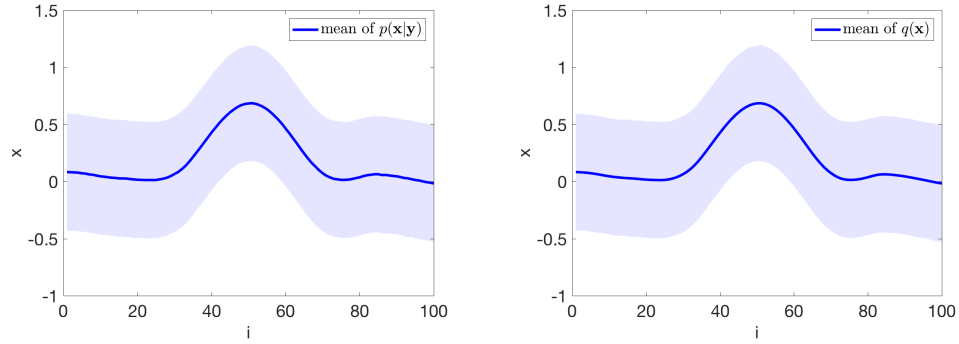
Despite the widespread use of variational type techniques in practice, the accuracy of the approximations is rarely theoretically studied. This has long been a challenging issue for approximate Bayesian inference, including the VGA. In this part, we conduct an experiment to numerically validate the VGA against the results by Markov chain Monte Carlo (MCMC). To this end, we employ the standard Metropolis-Hastings algorithm, with the Gaussian approximation from the VGA as the proposal distribution (i.e., independence sampler). In other words, we correct the samples drawn from VGA by a Metropolis-Hastings step. The length of the MCMC chain is 2×10^5, and the last 1×10^5 samples are used for computing the summarising statistics. From Fig. 2.7, one can observe that the trace plots do not show periodic pattern and the autocorrelations decay very fast, from which convergence of the chain can be concluded. The acceptance rate in the Metropolis-Hastings algorithm is 96.06%. This might be attributed to the fact that the VGA approximates the posterior distribution fairly accurately, and thus nearly all the proposals are accepted. The numerical results are

presented in Fig. 2.8, where the mean and the marginal 90% posterior credible intervals are shown, with the credible interval computed componentwise. It is observed that the mean and marginal posterior credible intervals by MCMC and VGA are very close to each other, cf. Figs. 2.8 and 2.9, thereby validating the accuracy of the VGA. The ℓ^2 error between the mean by MCMC and VGA is 9.80×10^{-3}, and the error between corresponding covariance in spectral norm is 6.40×10^{-3}. Graphically the means and covariances are indistinguishable, cf. Fig. 2.9.



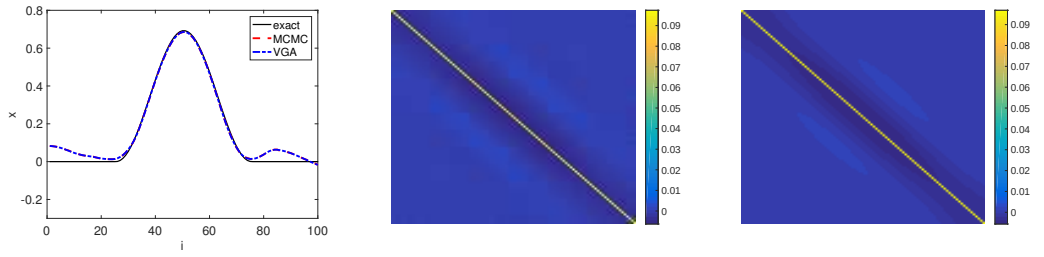
(a) the 20-th element (b) the 50-th element

Figure 2.7: Trace plots and autocorrelation of MCMC samples the 20-th and 50-th element.



(a) MCMC (b) VGA

Figure 2.8: The mean and marginal 90% posterior credible intervals by (a) MCMC and (b) VGA for `phillips` with $C_0 = 1.00 \times 10^{-1} \bar{C}_0$.

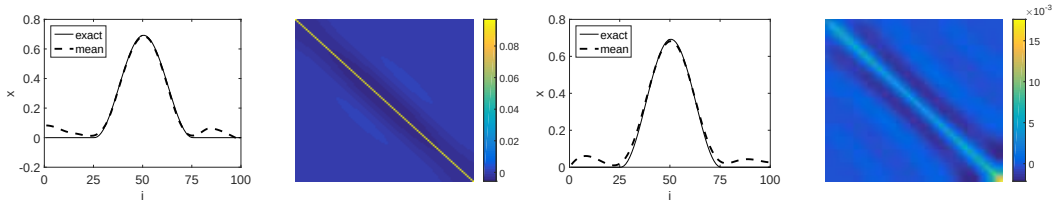


(a) mean (b) MCMC (c) VGA

Figure 2.9: (a) The mean by MCMC and VGA versus the exact solution, and the covariance by (b) MCMC and (c) VGA for `phillips` with $C_0 = 1.00 \times 10^{-1}\bar{C}_0$.

2.6.5 Numerical Reconstructions

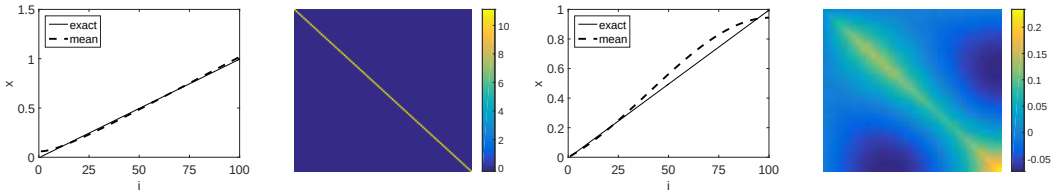
Last, we present VGAs for one- and two-dimensional examples. The numerical results for the following four 1D examples, i.e., `phillips`, `foxgood`, `gravity` and `heat`, for both L^2- and H^1-priors, are presented in Figs. 2.10-2.13. For the example `phillips` with either prior, the mean $\bar{x}$ by Algorithm 1 agrees very well with the true solution $x^\dagger$. However, near the boundary, the mean $\bar{x}$ is less accurate. This might be attributed to the fact that in these regions, the Poisson count is relatively small, and it may be insufficient for an accurate recovery. For the example `phillips`, the posterior mean $\bar{x}$ with H^1-prior is more smooth than that with the L^2-prior. This is due to the fact that H^1-prior penalises the gradient of the MAP estimate and VGA tends to fit around the mode. This difference caused by various priors again stresses the problem of model misspecification that careful choice of prior distribution is the first step towards a reasonable Bayesian exploration. For the L^2-prior, the optimal C is diagonal dominant, and decays rapidly away from the diagonal, cf. Fig. 2.10(b). For the H^1-prior, C remains largely diagonally dominant, but the off-diagonal entries decay a bit slower. Thus, it is valid to assume that C is dominated by local interactions as in Section 2.4.2. These observations remain largely valid for the other 1D examples, despite that they are much more ill-posed.



(a) $C_0 = 1.00 \times 10^{-1}\bar{C}_0$ (b) $C_0 = 2.5 \times 10^{-3}\bar{C}_1$

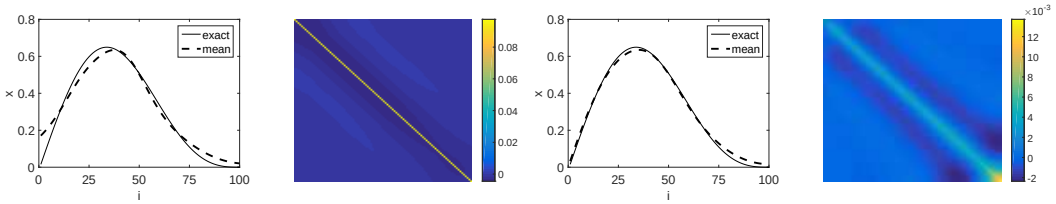
Figure 2.10: The Gaussian approximation for `phillips` shown with mean and covariance of the approximate distribution.

Last, we test Algorithm 1 on a 2D image of size 128×128, which takes 6473.16s on a MacBook Pro with 2.7 GHz Quad-Core Intel i7 CPU. In this example, the matrix $A \in \mathbb{R}^{16384\times 16384}$ is a (discrete) Gaussian blurring kernel with a blurring width 99, variance 1.5 and a circular boundary



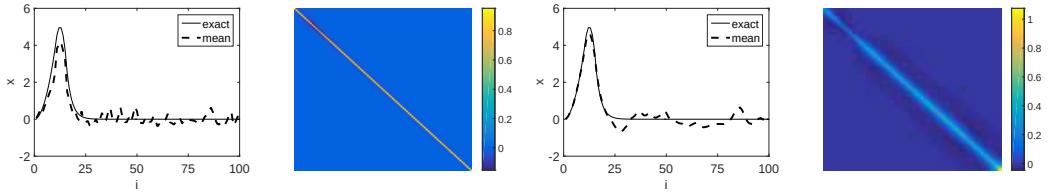
(a) $C_0 = 1.12 \times 10^{1} \bar{C}_0$ (b) $C_0 = 9.8 \times 10^{-3} \bar{C}_1$

Figure 2.11: The Gaussian approximation for `foxgood` shown with mean and covariance of the approximate distribution.



(a) $C_0 = 1 \times 10^{-1} \bar{C}_0$ (b) $C_0 = 1.5 \times 10^{-3} \bar{C}_1$

Figure 2.12: The Gaussian approximation for `gravity` shown with mean and covariance of the approximate distribution.



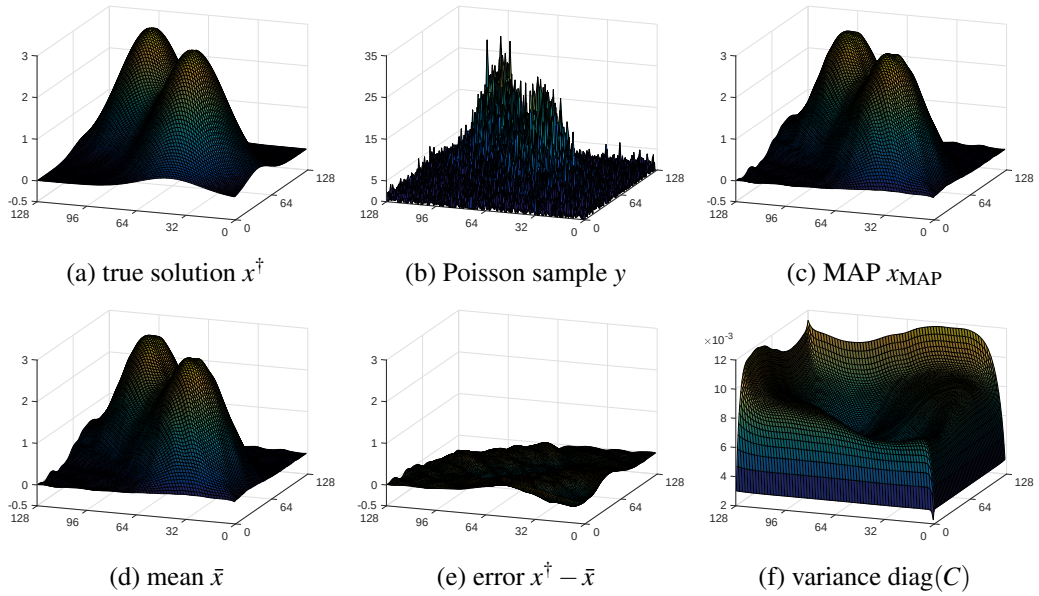
(a) $C_0 = 3.2 \times 10^{-1} \bar{C}_0$ (b) $C_0 = 1 \times 10^{0} \bar{C}_1$

Figure 2.13: The Gaussian approximation for `heat` shown with mean and covariance of the approximate distribution.

condition. Since the blurring width is large, the matrix A is indeed low-rank, and we employ a rSVD approximation of rank 2000, where the rank is determined by inspecting the singular value spectrum. The true solution $x^\dagger$ consists of two Gaussian blobs, cf. Fig. 2.14(a), and thus we employ a smooth prior with $C_0 = 6.00 \times 10^{-2} L^{-1} L^{-t}$, where $L = I \otimes L_1 + L_1 \otimes I$ is the 2D first-order finite difference matrix. Since the problem size is very large, we restrict C to be a sparse matrix such that every pixel interacts only with at most four neighbouring pixels. This allows reducing the computational cost greatly. The mean $\bar{x}$ is nearly identical with the true solution $x^\dagger$, and the error is very small, cf. Fig. 2.14. We also compare the mean $\bar{x}$ of the VGA solution with the MAP estimator $\hat{x}$ in three different measures, i.e., ℓ_2 error, structural similarity index and PSNR, which are 9.72, 0.812 and 18.64 for $\bar{x}$, respectively 9.74, 0.813, and 18.63 for $\hat{x}$. These results indicate that the mean $\bar{x}$ and the MAP estimator $\hat{x}$ represent equally good approximations. To indicate the uncertainty around the mean $\bar{x}$, we show in Fig. 2.14(f) the diagonal entries of C (i.e., the variance at each pixel). The variances are relatively large at pixels where the mean $\bar{x}$ is less accurate.

In summary, the VGA can provide a reliable point estimator together with useful covariance

estimates.



(a) true solution $x^\dagger$ (b) Poisson sample y (c) MAP x_{MAP}

(d) mean $\bar{x}$ (e) error $x^\dagger - \bar{x}$ (f) variance $\mathrm{diag}(C)$

Figure 2.14: The Gaussian approximation for image deblurring.

2.7 Conclusion

In this work, we have presented a study of the variational Gaussian approximation to the Poisson data (under the exponential inverse link function) with respect to the Kullback-Leibler divergence. We derived explicit expressions for the lower bound functional and its gradient, and proved its strict concavity and existence and uniqueness of an optimal Gaussian approximation. Then we developed an efficient algorithm for maximising the functional, discussed its convergence properties, and described practical strategies for reducing the complexity per iteration. Further, we analysed empirical Bayesian approach for automatically determining the hyperparameter using the variational Gaussian approximation, and proposed a monotonically convergent algorithm for the joint estimation. The numerical experiments indicate that the algorithm converges rapidly, and the variational Gaussian approximation can accurately capture the posterior distribution.

There are several avenues for further study. First, one of fundamental issues is the quality of the Gaussian approximation relative to the true posterior distribution. In general this issue has been long-standing, and it also remains to be analysed for the Poisson model. Second, the variational Gaussian approximation can be viewed as a nonstandard regularisation scheme, by also penalising the covariance. This naturally motivates the study on its regularising property from the perspective of classical regularisation theory, e.g., consistency and convergence rates. Third, the approach generally gives a very reasonable approximation. This suggests itself as a preconditioner for sampling techniques, e.g., variational approximation as the proposal distribution (i.e., independence sampler) in the standard Metropolis-Hastings type algorithm or as the base distribution for importance sam-

pler. It is expected to significantly speed up the convergence of these sampling procedures, which is confirmed by the preliminary experiments.

Chapter 3

Expectation Propagation for Poisson Data

3.1 Introduction

In the last chapter, we investigated how to leverage mathematical properties of the concrete problems, i.e., low rank structure of the forward operator and sparsity structure of the covariance matrix, to further improve variational inference. However, an important feature arising in many inverse problems, i.e., nonnegativity of the unknown variable, is not straightforward to incorporate. In this chapter, we investigate how to incorporate such constraints into approximate inference methods for a model related to emission tomography.

The maximum likelihood (ML) and maximum *a posteriori* (MAP) are currently the two most popular ways handling Poisson models in the literature [33, 44, 127, 89]. There are two major challenges in the development of numerical methods for Poisson data in the context of inverse problems and imaging, i.e., ill-posedness and nonnegativity constraints. To cope with the intrinsic ill-posed nature of the imaging problem, regularisation plays an important role: ML incorporates regularisation implicitly via early stopping during the iterative reconstruction, e.g., EM algorithm or Richardson-Lucy iterations, whereas MAP explicitly by imposing suitable penalties, e.g., Sobolev penalty, sparsity and total variation. Since the Poisson parameter has to be nonnegative to ensure the well-definedness of the Poisson likelihood, it naturally leads to nonnegativity constraints. Currently, there are two predominant ways dealing with the nonnegativity constraint. The first and most classical one is to reformulate the problem as a constrained optimisation problem with nonnegativity constraint on the signal vector, which can be then solved by algorithms like EM algorithm[125, 45] or gradient-based algorithms [75]. The second one is to approximate the Poisson distribution, e.g., using approximate Gaussian distributions for low count points [130] or using weighted quadratic surrogate. However, either ML or MAP estimate can only provide point estimates. Thus, the important issue of uncertainty quantification, which provides crucial reliability assessment on point estimates, is not fully addressed by these approaches.

In practice, a full Bayesian treatment of the posterior distribution is highly desirable [73, 128]. However, in the context of imaging, it is very challenging due to the nonnegativity constraint and

high-dimensionality of the imaging problem. In the machine learning literature, a number of approximate inference techniques have been proposed, e.g., variational inference [72, 20, 26, 14], expectation propagation [98, 99] and more recently Bayesian (deep) neural network [48]. In all these approaches, one aims at finding a best approximate yet tractable distribution within a family of parametric/nonparametric probability distributions (e.g., Gaussian or mixture of Gaussians), by minimising the error in a certain probability metric, prominently the Kullback-Leibler divergence. Empirically these methods can often produce reasonable estimate but at a much reduced computational cost. However, in these approaches, there seem no systematic strategies for handling constraints. For example, a straightforward truncation of the distribution due to the constraint often leads to elaborated distributions, e.g., truncated normal distribution, which tends to make the computation tedious or even intractable in variational Bayesian inference.

In this work, we develop an inference technique for Poisson data with constraints based on expectation propagation [98, 99], with a focus on Laplace type priors, to obtain an approximate Gaussian distribution. Laplace prior promotes the sparsity of the signal in a transformed domain, which is a valid assumption on natural images. The main contributions of the work are as follows. First, we present a unified treatment of two popular constraints in emission tomography within the framework of expectation propagation, by exploiting a separable form of the constraints. Second, we derive explicit update formulae in terms of one-dimensional integrals. It essentially exploits the rank-one projection form of the factors to reduce the intractable high-dimensional integrals to tractable one-dimensional ones. In this way, we arrive at two approximate inference algorithms, parameterised by either the moment or natural parameters. Third, we derive stable and efficient quadrature rules for evaluating the resulting one-dimensional integrals, i.e., a recursive scheme for Poisson sites with large counts and an approximate expansion for Laplace sites, and discuss different schemes for the recursion, dependent of the integral interval, in order to achieve good numerical stability. Last, we illustrate the potential of the approach with comprehensive numerical experiments with the posterior distribution formed by Poisson likelihood and an anisotropic total variation prior, including large-scale image tests and comparative study with MCMC and MAP estimates.

Last, we put the work in the context of Bayesian analysis of Poisson data. The only relevant work we are aware of is the recent work [82]. The work [82] discussed a full Bayesian exploration with EP, by modifying the posterior distributions using a rectified linear function on the transformed domain of the signal, which induces singular measures on the region violating the constraint. However, the work [82] does not consider the background. Note that expectation propagation has been applied to nonlinear inverse problems like electrical impedance tomography [52]; see also [70, 47] for related works on the variational Bayesian inference for inverse problems. However, none of these works discusses the case of Poisson data, and an approximate inference algorithm for Poisson data based on variational Bayes remains to be developed.

The rest of the chapter is organised as follows. In Section 3.2 we describe the posterior dis-

tribution for the Poisson likelihood function and Laplace type prior distribution. Then we give the explicit expressions of the integrals involved in expectation propagation and describe the algorithm in Section 3.3. In Section 3.4 we present stable and efficient numerical methods for evaluating one-dimensional integrals. Last, in Section 3.5 we present numerical results for one- and two-dimensional inverse problems. In Appendix B.1, we describe two useful parameterisations of a Gaussian distribution.

3.2 Problem Formulation

In this part, we give the Bayesian formulation for Poisson data, i.e., the likelihood function $p(y|x)$ and prior distribution $p(x)$, and discuss the nonnegativity constraint.

Let $x \in \mathbb{R}^n$ be the (unknown) signal/image of interest, $y \in \mathbb{R}_+^{m_1}$ be the observed Poisson data, and $A = [a_{ij}] = [a_i^t]_{i=1}^{m_1} \in \mathbb{R}_+^{m_1 \times n}$ be the forward map, where the superscript t denotes matrix / vector transpose. The entries of the matrix A are assumed to be nonnegative. For example, in emission computed tomography, it can be a discrete analogue of Radon transform, or probabilistically, the entry a_{ij} of the matrix A denotes the probability that the ith sensor pair records the photon emitted at the jth site.

The conditional probability density $p(y_i|x)$ of observing $y_i \in \mathbb{N}$ given the signal x is given by

$$p(y_i|x) = \frac{(a_i^t x + r_i)^{y_i} e^{-(a_i^t x + r_i)}}{y_i!},$$

where $r = [r_i]_i \in \mathbb{R}_+^{m_1}$ is the background. That is, the entry y_i follows a Poisson distribution with a parameter $a_i^t x + r_i$. The Poisson model of this form is popular in the statistical modelling of inverse and imaging problems involving counts, e.g., positron emission tomography [131]. If the entries of y are independent and identically distributed (i.i.d.), then the likelihood function $p(y|x)$ is given by

$$p(y|x) = \prod_{i=1}^{m_1} p(y_i|x). \tag{3.1}$$

Note that the vanilla likelihood function $p(y|x)$ is not well-defined for all $x \in \mathbb{R}^n$, and suitable constraints on x are needed in order to ensure the well-definedness of the factors $p(y_i|x)$'s. In the literature, there are three popular constraints:

C1: $x \in \mathscr{C}_1 = \{x|x > 0\} := \cap_i \{x|x_i > 0\}$;

C2: $x \in \mathscr{C}_2 = \{x|Ax > 0\} := \cap_i \{x|[Ax]_i = a_i^t x > 0\}$;

C3: $x \in \mathscr{C}_3 = \{x|Ax + r > 0\} := \cap_i \{x|[Ax + r]_i = a_i^t x + r_i > 0\}$.

Since the entries of A are nonnegative, there holds $\mathscr{C}_1 \subset \mathscr{C}_2 \subset \mathscr{C}_3$. In practice, the first assumption is most consistent with the physics in that it reflects the physical constraint that emission counts are non-negative. The last two assumptions were proposed to reduce positive bias in the cold region

[89], i.e., the region that has zero count. In this work, we shall focus on the last two constraints. Note that relaxed versions of C1 and C2, where $>$ is relaxed to be $\geq$, could allow zero intensity in the domain and thus are more suitable for problems like PET. And one can extend our discussion to the settings with the equality included by modifying the Poisson likelihood functions.

The constraints C2 and C3 can be unified, which is useful for the discussions below.

Definition 3.2.1. *For each likelihood factor $p(y_i|x)$ with the constraint C2, let*

$$V_i^+ = \{x|[Ax]_i = a_i^t x > 0\} \quad \textit{and} \quad V_i^- = \mathbb{R}^n \backslash V_i^+.$$

For each likelihood factor $p(y_i|x)$ with the constraint C3, let

$$V_i^+ = \{x|[Ax+r]_i = a_i^t x + r_i > 0\} \quad \textit{and} \quad V_i^- = \mathbb{R}^n \backslash V_i^+. \tag{3.2}$$

Then the constraints C2 and C3 are both given by $V^+ = \cap_i V_i^+$ and $V^- = \mathbb{R}^n \backslash V^+$.

With the indicator function $\mathbf{1}_{V^+}(x)$ of the set V^+, we modify the likelihood function $p(y|x)$ by

$$\ell(x) = p(y|x)\mathbf{1}_{V^+}(x). \tag{3.3}$$

This extends the domain of the likelihood function $p(y|x)$ from V^+ to $\mathbb{R}^n$, and it facilitates a full Bayesian treatment. Since the indicator function $\mathbf{1}_{V^+}(x)$ admits a separable form, i.e., $\mathbf{1}_{V^+}(x) = \prod_{i=1}^{m_1} \mathbf{1}_{V_i^+}(x)$, the modified likelihood function $\ell(x)$ factorises into

$$\ell(x) = \prod_{i=1}^{m_1} \ell_i(x) \quad \text{with } \ell_i(x) = p(y_i|x)\mathbf{1}_{V_i^+}(x). \tag{3.4}$$

To fully specify the Bayesian model, we have to stipulate the prior distribution $p(x)$. In this work, we focus on an anisotropic total variation prior, but describe the approach for a general Laplace type prior. Let $L \in \mathbb{R}^{m_2 \times n}$ and $L_i^t \in \mathbb{R}^{n \times 1}$ be the ith row of L. Then a general Laplace type prior $p(x)$ is given by

$$p(x) = \prod_{i=1}^{m_2} p_i(x) \quad \text{with } p_i(x) = \tfrac{\mu}{2} e^{-\mu|L_i^t x|}. \tag{3.5}$$

The parameter $\mu > 0$ determines the strength of the prior, playing the crucial role of regularisation parameter in variational regularisation [67]. The prior $p(x)$ is commonly known as a sparsity prior (in the transformed domain), which favours a candidate with many small elements and few large elements in the vector Lx. The canonical (anisotropic) total variation prior is recovered when the matrix L computes the discrete gradient. It is well known that the total variation penalty can preserve well edges in the image/signals, and hence it has been very popular for various imaging tasks, e.g., denoising, deblurring and superresolution [121, 27]. Note that the total variation prior is also an improper prior due to the same reason discussed in Remark 2.6.1. Thanks to the fact that the null

space of forward operator A does not have non-zero adjoint with that of the difference operator L, the joint distribution and tilted distributions are well-defined and the EP framework is thus applicable.

By Bayes' formula, we obtain the Bayesian solution to the Poisson inverse problem:

$$p(x|y) = Z^{-1}(y) \prod_{i=1}^{m_1} \ell_i(x) \prod_{i=1}^{m_2} p_i(x), \tag{3.6}$$

where $Z(y)$ is the normalising constant, defined by

$$Z(y) = \int_{\mathbb{R}^n} \prod_{i=1}^{m_1} \ell_i(x) \prod_{i=1}^{m_2} p_i(x) \mathrm{d}x.$$

The computation of $Z(y)$ is generally intractable for high-dimensional problems, and the posterior distribution $p(x|y)$ has to be approximated. Note that for a single reconstruction problem, the observation y is fixed. In the following, we would omit y in Z and other factors in the posterior distribution for notation simplicity.

3.3 Approximate Inference by Expectation Propagation

In this section, we describe the basic concepts and algorithms of expectation propagation (EP), for exploring the posterior distribution for the Poisson data with a Laplace type prior. EP due to Minka [98, 99] is a popular variational type approximate inference method in the machine learning literature. It is especially suitable for approximating a distribution formed by a product of functions, with each factor being of projection form. Since its first appearance in 2001, EP has found many successful applications in practice, and it is reported to be very accurate, e.g., for Gaussian processes [112], and electrical impedance tomography with sparsity prior [52]. Despite numerous empirical successes, the theoretical understanding of EP remains quite limited; see the works [34, 35] for recent progress.

EP looks for an approximate Gaussian distribution $q(x)$ to the target distribution $p(x)$ by means of an iterative algorithm. It exploits essentially the following factorization of the posterior distribution $p(x|y)$ (with $m = m_1 + m_2$ being the total number of factors):

$$p(x|y) = Z^{-1} \prod_{i=1}^{m} t_i(x), \quad \text{with } t_i(x) = \begin{cases} \ell_i(x), & i = 1, \ldots, m_1, \\ p_{i-m_1}(x), & i = m_1 + 1, \ldots, m. \end{cases} \tag{3.7}$$

Note that each factor $t_i(x)$ is a function defined on the whole space $\mathbb{R}^n$. Likewise, we denote the Gaussian approximation $q(x)$ to the posterior distribution $p(x|y)$ by

$$q(x) = \tilde{Z}^{-1} \prod_{i=1}^{m} \tilde{t}_i(x),$$

with each factor $\tilde{t}_i(x)$ being a Gaussian distribution $\mathcal{N}(x|\mu_i, C_i)$, and $\tilde{Z}$ is the corresponding nor-

malizing constant. Below we use two different parameterisations of a Gaussian distribution, i.e., moment parameters (mean and covariance) (μ, C) and natural parameters (h, Λ); see Appendix B.1 for details.

3.3.1 Reduction to One-dimensional Integrals

There are two main steps of one EP iteration: (a) forming a tilted distribution $\hat{q}_i(x)$, and (b) updating the Gaussian approximation $q(x)$ by matching its moments with that of $\hat{q}_i(x)$. The moment matching step can be interpreted as minimising Kullback-Leilber divergence $\mathrm{KL}(\hat{q}_i||q)$ [98, 99, 52].

The task at step (a) is to construct the ith tilted distribution $\hat{q}_i(x)$. Let $q_{\setminus i}(x)$ be the ith cavity distribution, i.e., the product of all but the ith factor, and defined by

$$q_{\setminus i}(x) = Z_i^{-1} \prod_{j \neq i} \tilde{t}_i(x) \tag{3.8}$$

with the normalising constant $Z_i = \int_{\mathbb{R}^n} \prod_{j \neq i} \tilde{t}_i(x) \mathrm{d}x$. The cavity distribution $q_{\setminus i}(x)$ is Gaussian, i.e., $q_{\setminus i}(x) = \mathcal{N}(x|\mu_{\setminus i}, C_{\setminus i})$. Then the ith tilted distribution $\hat{q}_i(x)$ of the approximation $q(x)$ is given by

$$\hat{q}_i(x) = \hat{Z}_i^{-1} t_i(x) \prod_{j \neq i} \tilde{t}_i(x), \tag{3.9}$$

where $\hat{Z}_i = \int_{\mathbb{R}^n} t_i(x) \prod_{j \neq i} \tilde{t}_i(x) \mathrm{d}x$ is the corresponding normalising constant. With the exclusion-inclusion step, one replaces the ith factor $\tilde{t}_i(x)$ in the approximation q with the exact one $t_i(x)$.

The task at step (b) is to compute moments of the ith tilde distribution $\hat{q}_i(x)$, which are then used to update the approximation $q(x)$. This requires integration over $\mathbb{R}^n$, which is generally numerically intractable, if $\hat{q}_i(x)$ were arbitrary. Fortunately, each factor $t_i(x)$ in the factorisation (3.7) is of projection form and depends only on the scalar $u^t x$, with $u \in \mathbb{R}^n$. This is the key fact to render relevant high-dimensional integrals numerically tractable. Below we write the factor $t_i(x)$ as $\bar{t}_i(u_i^t x)$ and accordingly, the ith cavity function $\hat{q}_i(x)$ as

$$\hat{q}_i(x) = \hat{Z}_i^{-1} \bar{t}_i(u_i^t x) \mathcal{N}(x|\mu_{\setminus i}, C_{\setminus i}), \tag{3.10}$$

upon replacing $\prod_{j \neq i} \tilde{t}_i(x)$ with its normalised version $\mathcal{N}(x|\mu_{\setminus i}, C_{\setminus i})$, and changing the normalising constant $\hat{Z}_i$ accordingly.

Since a Gaussian distribution is fully determined by the mean and covariance, it suffices to evaluate the first three (0th to 2nd) moments of $\hat{q}_i(x)$. The projection form in the factor t_i allows reducing the moment evaluation of $\hat{q}_i(x)$ to 1D integrals. Theorem 3.3.1 gives the explicit update scheme for the Gaussian approximation $q(x)$ from the cavity distribution $q_{\setminus i}(x)$, whose moment and natural parameters are denoted by $(\mu_{\setminus i}, C_{\setminus i})$ and $(h_{\setminus i}, \Lambda_{\setminus i})$, respectively.

Theorem 3.3.1. *The normalising constant* $\hat{Z}_i := \int_{\mathbb{R}^n} \bar{t}_i(u_i^t x)\mathcal{N}(x|\mu_{\setminus i}, C_{\setminus i})\mathrm{d}x$ *is given by*

$$\hat{Z}_i = \int_{\mathbb{R}} \bar{t}_i(s)\mathcal{N}(s|u_i^t\mu_{\setminus i}, u_i^t C_{\setminus i}u_i)\mathrm{d}s =: Z_s$$

Then with the auxiliary variables $\bar{s} \in \mathbb{R}$ *and* C_s *defined by*

$$\bar{s} = Z_s^{-1}\int_{\mathbb{R}} \bar{t}_i(s)\mathcal{N}(s|u_i^t\mu_{\setminus i}, u_i^t C_{\setminus i}u_i)s\mathrm{d}s \quad \textit{and} \quad C_s = Z_s^{-1}\int_{\mathbb{R}} \bar{t}_i(s)\mathcal{N}(s|u_i^t\mu_{\setminus i}, u_i^t C_{\setminus i}u_i)s^2\mathrm{d}s - \bar{s}^2, \tag{3.11}$$

the mean $\mu = \mathbb{E}_{\hat{q}_i}[x]$ *and covariance* $C = \mathbb{V}_{\hat{q}_i}[x]$ *are given respectively by*

$$\begin{aligned}
\mu &= \mu_{\setminus i} + C_{\setminus i}u_i(u_i^t C_{\setminus i}u_i)^{-1}(\bar{s} - u_i^t\mu_{\setminus i}),\\
C &= C_{\setminus i} + (u_i^t C_{\setminus i}u_i)^{-2}(C_s - u_i^t C_{\setminus i}u_i)C_{\setminus i}u_i u_i^t C_{\setminus i}.
\end{aligned}$$

Similarly, the precision mean $h_{\hat{q}_i}$ *and precision* $\Lambda_{\hat{q}_i}$ *are given respectively by*

$$\begin{aligned}
h_{\hat{q}_i} &= h_{\setminus i} + \lambda_{1,i}u_i \quad \textit{with } \lambda_{1,i} = \frac{\bar{s}}{C_s} - \frac{u_i^t\mu_{\setminus i}}{u_i^t C_{\setminus i}u_i},\\
\Lambda_{\hat{q}_i} &= \Lambda_{\setminus i} + \lambda_{2,i}u_i u_i^t \quad \textit{with } \lambda_{2,i} = \frac{1}{C_s} - \frac{1}{u_i^t C_{\setminus i}u_i}.
\end{aligned}$$

Proof. The expressions for $\hat{Z}_i$, μ and C were given in [52, Section 3]. Thus it suffices to derive the formulae for (h, Λ). Recall Sherman-Morrison-Woodbury formula [54, p. 65]: for any invertible $B \in \mathbb{R}^{n\times n}$, $u, v \in \mathbb{R}^n$, there holds

$$(B + uv^t)^{-1} = B^{-1} - \frac{B^{-1}uv^t B^{-1}}{1 + v^t B^{-1}u}. \tag{3.12}$$

Let $\lambda = (u_i^t C_{\setminus i}u_i)^{-2}(C_s - u_i^t C_{\setminus i}u_i)$. Then the precision matrix Λ is given by

$$\begin{aligned}
\Lambda &= (C_{\setminus i} + C_{\setminus i}u_i\lambda u_i^t C_{\setminus i})^{-1}\\
&= C_{\setminus i}^{-1} - u_i(\lambda^{-1} + u_i^t C_{\setminus i}u_i)^{-1}u_i^t\\
&= \Lambda_{\setminus i} + \Big(\frac{1}{C_s} - \frac{1}{u_i^t C_{\setminus i}u_i}\Big)u_i u_i^t.
\end{aligned}$$

Similarly, the precision mean $h := \Lambda\mu$ is given by

$$\begin{aligned}
h &= \Big[\Lambda_{\setminus i} + \Big(\frac{1}{C_s} - \frac{1}{u_i^t C_{\setminus i}u_i}\Big)u_i u_i^t\Big][\mu_{\setminus i} + C_{\setminus i}u_i(u_i^t C_{\setminus i}u_i)^{-1}(\bar{s} - u_i^t\mu_{\setminus i})]\\
&= \Lambda_{\setminus i}\mu_{\setminus i} + u_i\Big(\frac{\bar{s}}{C_s} - \frac{u_i^t\mu_{\setminus i}}{u_i^t C_{\setminus i}u_i}\Big) = h_{\setminus i} + u_i\Big(\frac{\bar{s}}{C_s} - \frac{u_i^t\mu_{\setminus i}}{u_i^t C_{\setminus i}u_i}\Big).
\end{aligned}$$

This completes the proof of the theorem. □

In both approaches, the 1D integrals $(Z_s, \bar{s}, C_s)$ are needed, which depend on $u_i^t \mu_{\setminus i}$ and $u_i^t C_{\setminus i} u_i$. A direct approach is first to downdate (the Cholesky factor of) Λ and then to solve a linear system. In practice, this can be expensive and the cost can be mitigated. Indeed, they can be computed without the downdating step; see Lemma 3.3.1 below. This fact was implicitly stated in [52]. Both approaches have their pros and cons: the moment one does not require solving linear systems, and the natural one allows singular Gaussians for $\tilde{t}_i(x)$. Below we use the super- or subscript n and o to denote a variable updated at current iteration from that of the last iteration.

Lemma 3.3.1. *Let $c = u_i^t \Lambda_o^{-1} u_i = u_i^t C_o u_i$, (h, Λ) be the natural parameter of $q(x)$ and $(\lambda_{1,i}, \lambda_{2,i})$ be defined in Theorem 3.3.1. Then the mean $u_i^t \mu_{\setminus i}$ and variance $u_i^t C_{\setminus i} u_i$ of the Gaussian distribution $\mathcal{N}(s | u_i^t \mu_{\setminus i}, u_i^t C_{\setminus i} u_i)$ are respectively given by*

$$u_i^t \mu_{\setminus i} = \frac{u_i^t \Lambda_o^{-1} h - c\lambda_{1,i}^o}{1 - c\lambda_{2,i}^o} \quad \text{and} \quad u_i^t C_{\setminus i} u_i = \frac{c}{1 - c\lambda_{2,i}^o}. \tag{3.13}$$

Proof. We suppress the sub/superscript o. By the definition of $u_i^t C_{\setminus i} u_i$ and (3.12), we have

$$\begin{aligned} u_i^t C_{\setminus i} u_i &= u_i^t (\Lambda - \lambda_{2,i} u_i u_i^t)^{-1} u_i \\ &= u_i^t [\Lambda^{-1} - \Lambda^{-1} u_i (-\lambda_{2,i}^{-1} + c)^{-1} u_i^t \Lambda^{-1}] u_i \\ &= c - c(-\lambda_{2,i}^{-1} + c)^{-1} c = \frac{c}{1 - c\lambda_{2,i}}, \end{aligned}$$

and similarly, we have

$$\begin{aligned} u_i^t \mu_{\setminus i} &= u_i^t (\Lambda - \lambda_{2,i} u_i u_i^t)^{-1} (h - \lambda_{1,i} u_i) \\ &= u_i^t [\Lambda^{-1} - \Lambda^{-1} u_i (-\lambda_{2,i}^{-1} + c)^{-1} u_i^t \Lambda^{-1}](h - \lambda_{1,i} u_i) = \frac{u_i^t \Lambda^{-1} h - c\lambda_{1,i}}{1 - c\lambda_{2,i}}. \end{aligned}$$

This completes the proof of the lemma. □

Since the quantities for the 1D integrals can be calculated from variables updated in the last iteration, it is unnecessary to form cavity distributions. Indeed, the cavity precision is formed by $\Lambda_{\setminus i} = \Lambda_o - \lambda_{2,i}^o u_i u_i^t$, and the updated precision is given by $\Lambda_n = \Lambda_{\setminus i} + \lambda_{2,i}^n u_i u_i^t$; and similarly for h. Thus, we can update Λ directly with $(\lambda_{2,i}^o, \lambda_{2,i}^n)$ and h with $(\lambda_{1,i}^o, \lambda_{1,i}^n)$; this is summarised in the next remark.

Remark 3.3.1. *The differences $\lambda_{k,i}^n - \lambda_{k,i}^o$, $k = 1, 2$, can be used to update the natural parameter (h, Λ):*

$$\lambda_{1,i}^n - \lambda_{1,i}^o = \frac{\bar{s}}{C_s} - \frac{u_i^t \mu_o}{u_i^t C_o u_i} \quad \text{and} \quad \lambda_{2,i}^n - \lambda_{2,i}^o = \frac{1}{C_s} - \frac{1}{u_i^t C_o u_i}. \tag{3.14}$$

Moreover, the sign of $\lambda_{2,i}^n - \lambda_{2,i}^o$ determines whether to update or downdate the Cholesky factor of Λ. This will be adopted in the implementation of EP algorithms.

3.3.2 Update Schemes and Algorithms

Now we can state the direct update scheme, i.e. without explicitly constructing the medium cavity distribution $q_{\setminus i}(x)$, for both natural and moment parameterisations.

Theorem 3.3.2. *Let (h,Λ) and (μ,C) be the natural and moment parameters of the Gaussian approximation $q(x)$, respectively. The following update schemes hold.*

(i) *The precision mean h and precision Λ can be updated by*

$$h_n = h_o + \Big(\frac{\bar{s}}{C_s} - \frac{u_i^t\Lambda_o^{-1}h_o}{u_i^t\Lambda_o^{-1}u_i}\Big)u_i \quad \textit{and} \quad \Lambda_n = \Lambda_o + \Big(\frac{1}{C_s} - \frac{1}{u_i^t\Lambda_o^{-1}u_i}\Big)u_iu_i^t. \tag{3.15}$$

(ii) *The mean μ and covariance C can be updated by*

$$\mu_n = \mu_o + \frac{\bar{s}-u_i^t\mu_o}{u_i^tC_ou_i}C_ou_i \quad \textit{and} \quad C_n = C_o + \Big(\frac{C_s}{(u_i^tC_ou_i)^2} - \frac{1}{u_i^tC_ou_i}\Big)(C_ou_i)(u_i^tC_o). \tag{3.16}$$

Proof. The first assertion is direct from Theorem 3.3.1 and Remark 3.3.1, and it can be rewritten as

$$\Lambda_n = \Lambda_o + (\lambda_{2,i}^n - \lambda_{2,i}^o)u_iu_i^t \quad \text{and} \quad h_n = h_o + (\lambda_{1,i}^n - \lambda_{1,i}^o)u_i. \tag{3.17}$$

By Sherman-Morrison-Woodbury formula (3.12), the covariance $C_n = \Lambda_n^{-1}$ is given by

$$\begin{aligned} C_n &= (\Lambda_o + (\lambda_{2,i}^n - \lambda_{2,i}^o)u_iu_i^t)^{-1} \\ &= \Lambda_o^{-1} - \Lambda_o^{-1}u_i\Big(\frac{1}{\lambda_{2,i}^n - \lambda_{2,i}^o} + u_i^tC_ou_i\Big)^{-1}u_i^t\Lambda_o^{-1} \\ &=: C_o + \eta_2(C_ou_i)(u_i^tC_o), \end{aligned}$$

where the scalar $\eta_2 := -(\frac{1}{\lambda_{2,i}^n - \lambda_{2,i}^o} + u_i^tC_ou_i)^{-1}$ can be simplified to

$$\eta_2 = -\frac{\lambda_{2,i}^n - \lambda_{2,i}^o}{1 + (\lambda_{2,i}^n - \lambda_{2,i}^o)u_i^tC_ou_i} = -\frac{1}{u_i^tC_ou_i} + \frac{C_s}{(u_i^tC_ou_i)^2},$$

where the second identity line follows from Remark 3.3.1. Similarly, the mean $\mu_n := C_nh_n$ is given by

$$\begin{aligned} \mu_n &= [C_o + \eta_2(C_ou_i)(u_i^tC_o)][h_o + (\lambda_{1,i}^n - \lambda_{1,i}^o)u_i] \\ &= \mu_o + (\lambda_{1,i}^n - \lambda_{1,i}^o)C_ou_i + \eta_2u_i^t\mu_oC_ou_i + \eta_2(\lambda_{1,i}^n - \lambda_{1,i}^o)u_i^tC_ou_iC_ou_i =: \mu_o + \eta_1C_ou_i, \end{aligned}$$

where the scalar $\eta_1 = (\lambda_{1,i}^n - \lambda_{1,i}^o) + \eta_2u_i^t\mu_o + \eta_2(\lambda_{1,i}^n - \lambda_{1,i}^o)u_i^tC_ou_i$ can be simplified to

$$\eta_1 = \frac{(\lambda_{1,i}^n - \lambda_{1,i}^o) - (\lambda_{2,i}^n - \lambda_{2,i}^o)u_i^t\mu_o}{1 + (\lambda_{2,i}^n - \lambda_{2,i}^o)u_i^tC_ou_i} = \frac{\bar{s}-u_i^t\mu_o}{u_i^tC_ou_i},$$

where the second identity is due to Remark 3.3.1. □

All matrix operations in Theorem 3.3.2 are of rank one type, which can be implemented stably and efficiently with the Cholesky factors and their update / downdate; see Section 3.3.3 for details. Thus, in practice, we employ Cholesky factors of the precision Λ and covariance C (of $q(x)$), denoted by Λ_{chol} and C_{chol}, respectively, instead of Λ and C directly. Further, we also use the auxiliary variables $(\lambda_{1,i}, \lambda_{2,i})$, and stack $\{(\lambda_{1,i}, \lambda_{2,i})\}_{i=1}^{m_1+m_2}$ into two vectors

$$\lambda_1 = [\lambda_{1,i}]_i, \quad \lambda_2 = [\lambda_{2,i}]_i \in \mathbb{R}^{m_1+m_2},$$

which are initialised to zeros. In summary, we obtain two approximate inference procedures for Poisson data with a Laplace type prior; see Algorithms 3 and 4 for details. The important practical task of computing the 1D integrals in Theorem 3.3.1 will be discussed in Section 3.4 below in detail.

The rigorous convergence analysis of EP is an outstanding issue. Nonetheless, empirically, it often converges very fast, which is also observed in our numerical experiments in Section 3.5. In practice, one can terminate the iteration by monitoring the relative change of the parameters or fixing the maximum number K of iterations. The optimal choice of the hyperparameter μ in the prior $p(x)$ is notoriously challenging [67]. One may apply hierarchical Bayesian modelling in order to estimate it from the data simultaneously with $q(x)$ [136, 71, 14]. However, we shall not delve into the issue further.

Algorithm 3 Expectation propagation for Poisson data (natural parametrisation)

1: Input: (A, y), hyper-parameter μ, and maximum number K of iterations
2: Initialise h, Λ_{chol}, λ_1 and λ_2;
3: **for** $k = 1, 2, \ldots, K$ **do**
4: Randomly choose an index i to update;
5: Compute the mean and variance for 1D Gaussian integral by Lemma 3.3.1;
6: Evaluate $\bar{s}$ and C_s in (3.11);
7: Calculate and update $\lambda_{1,i}$ and $\lambda_{2,i}$;
8: Update h and Λ_{chol} by Theorem 3.3.2;
9: Check the stopping criterion.
10: **end for**
11: Output: (h, Λ_{chol})

3.3.3 Efficient Implementation and Complexity Estimate

The rank-one matrix update $A \pm \alpha uu^t$, for $A \in \mathbb{R}^{n\times n}$, $u \in \mathbb{R}^n$ and $\alpha > 0$, can be stably and efficiently updated / downdated with the Cholesky factor of A with $\sqrt{\alpha}u$. The update step of A can be viewed as an iteration from A_k to A_{k+1}. Let the upper triangular matrices R_k and R_{k+1} be the Cholesky factors of A_k and A_{k+1} respectively, i.e., $A_k = R_k^t R_k$ and $A_{k+1} = R_{k+1}^t R_{k+1}$. There are two possible cases:

(i) If $A_{k+1} = A_k + \alpha uu^t$, R_{k+1} is the Cholesky rank one update of R_k with $\sqrt{\alpha}u$.

(ii) If $A_{k+1} = A_k - \alpha uu^t$, R_{k+1} is the Cholesky rank one downdate of R_k with $\sqrt{\alpha}u$.

Algorithm 4 Expectation propagation for Poisson data (moment parametrisation)

1: Input: (A, y), hyper-parameter μ, and maximum number K of iterations
2: Initialise μ, C_{chol}, λ_1 and λ_2;
3: **for** $k = 1, 2, \ldots, K$ **do**
4: Randomly choose an index i to update;
5: Compute the mean and variance for 1D Gaussian integral by Lemma 3.3.1;
6: Evaluate $\bar{s}$ and C_s in (3.11);
7: Calculate and update $\lambda_{1,i}$ and $\lambda_{2,i}$;
8: Update μ and C_{chol} by Theorem 3.3.2;
9: Check the stopping criterion.
10: **end for**
11: Output: (μ, C_{chol})

The update/downdate is available in several packages. For example, in `MATLAB`, the function `cholupdate` implements the update/downdate of Cholesky factors, based on `LAPACK` subroutines `ZCHUD` and `ZCHDD`.

Next, we analyse the computational complexity per inner iteration. The first step in each iteration picks up one index i, which is of constant complexity. For the second step, i.e., computing the mean and variance for 1D integrals, the dominant part is linear solve involving upper triangular matrices and matrix-vector product for natural and moment parameters. For either parameterisation, it incurs $\mathcal{O}(n^2)$ operations. The third step computes $\bar{s}$ and C_s from the one dimensional integrals. For Poisson site, the complexity is $\mathcal{O}(y_i)$, and for Laplace site, it is $\mathcal{O}(1)$. Last, the fourth step is dominated by Cholesky factor modifications, and its complexity is $\mathcal{O}(n^2)$. Overall, the computational complexity per inner iteration is $\mathcal{O}(n^2 + y_i)$. In a large data setting, $y_i \ll n$, and thus the complexity is about $\mathcal{O}(n^2)$.

In passing, we note that in practice, the covariance / precision matrix may admit additional structures, e.g., sparsity, which translate into structures on the corresponding Cholesky factors. For the general sparsity assumption, it seems unclear how to effectively exploit it for Cholesky update/downdate for enhanced efficiency, except the diagonal case, which can be incorporated into the algorithm straightforwardly.

3.4 Stable Evaluation of 1d Integrals

In this section, we develop a stable implementation for the three 1D integrals: Z_s, $\bar{s}$ and C_s in Theorem 3.3.1. These integrals form the basic components of Algorithms 3 and 4, and their stable, accurate and efficient evaluation is crucial to the performance of the algorithms. By suppressing the subscript i, we can write the integrals in a unified way:

$$J_j = \int_{\mathbb{R}} \bar{t}(s) \mathcal{N}(s|m, \sigma^2) s^j \mathrm{d}s, \quad j = 0, 1, 2,$$

where the factor $\bar{t}(s)$ is either the constrained Poisson likelihood or the Laplace prior. Then we can express $\bar{s}$ and C_s in terms of J_j by

$$\bar{s} = \frac{J_1}{J_0} \quad \text{and} \quad C_s = \frac{J_2}{J_0} - \bar{s}^2. \tag{3.18}$$

Note that the normalising constants in J_j cancel out in $\bar{s}$ and C_s, and thus they can be ignored when evaluating the integrals. Below we derive the formulae for the constrained Poisson likelihood and Laplace prior separately. In essence, the computation boils down to stable evaluation of moments of a (truncated) Gaussian distribution. This task was studied in several works [32, 124]: [32] focuses on Gaussian moments, and [124] discusses also evaluating the integrals involving Laplace distributions. In this work, we discuss moments involving both Laplace and Poisson distributions.

3.4.1 Constrained Poisson Likelihood

Throughout, we suppress the subscript i, write V_+ etc in place V_i^+ etc and introduce the scaler variable $s = a^t x$. Then the constraint on x transfers to that on s: $a^t x > 0$ corresponds to $s > 0$ and $a^t x + r > 0$ to $s > -r$, respectively. We shall slightly abuse the notation and use $\mathbf{1}_{V_+}(s)$ as the indicator for the constraint on s. Then the Poisson likelihood $t(x)$ can be equivalently written in either x or s as

$$t(x) = \frac{(a^t x + r)^y e^{-(a^t x + r)}}{y!} \mathbf{1}_{V_+}(x) \quad \text{and} \quad \bar{t}(s) = \frac{(s+r)^y e^{-(s+r)}}{y!} \mathbf{1}_{V_+}(s). \tag{3.19}$$

Note that the factorial $y!$ cancels out when computing $\bar{s}$ and C_s, so it is omitted in the derivation below. For a fixed $\mathcal{N}(s|m,\sigma^2)$, the integrals $J_{y,j}$ depend on the observed count data y and moment order j:

$$\begin{aligned} J_{y,j} &= \int_{\mathbb{R}} (s+r)^y e^{-(s+r)} \mathbf{1}_{V_+}(s) \mathcal{N}(s|m,\sigma^2) s^j \mathrm{d}s \\ &= \int_b^\infty (s+r)^y s^j e^{-(s+r)} \mathcal{N}(s|m,\sigma^2) \mathrm{d}s. \end{aligned} \tag{3.20}$$

where the lower integral bound $b = 0$ or $b = -r$, which is evident from the context. Note that the terms $e^{-(s+r)}$ and $\mathcal{N}(s|m,\sigma^2)$ in $J_{y,j}$ together give an unnormalised Gaussian density. This allows us to reduce the integrals $J_{y,j}$ into (truncated) Gaussian moment evaluations of the type:

$$I_y = \int_b^\infty (s+r)^y \mathcal{N}(s|m-\sigma^2,\sigma^2) \mathrm{d}s, \tag{3.21}$$

and accordingly $\bar{s}$ and C_s. This is given in the next result.

Now we can express $\bar{s}$ and C_s in terms of I_y.

Theorem 3.4.1. *$\bar{s}$ and C_s can be computed by*

$$\bar{s} = \frac{I_{y+1}}{I_y} - r \quad \textit{and} \quad C_s = \frac{I_{y+2}}{I_y} - \Big(\frac{I_{y+1}}{I_y}\Big)^2. \tag{3.22}$$

Proof. First, we claim that with $\alpha = e^{\frac{\sigma^2}{2}-m-r}$, there hold the following identities

$$J_{y,0} = \alpha I_y, \quad J_{y,1} = \alpha(I_{y+1} - rI_y), \quad \text{and} \quad J_{y,2} = \alpha(I_{y+2} - 2rI_{y+1} + r^2 I_y). \tag{3.23}$$

Let $c_\sigma = (2\pi\sigma^2)^{-\frac{1}{2}}$. Then by completing the squares we obtain

$$e^{-(s+r)}\mathcal{N}(s|m,\sigma^2) = c_\sigma e^{-r-s-\frac{(s-m)^2}{2\sigma^2}} = c_\sigma e^{\frac{\sigma^2}{2}-m-r} e^{-\frac{(s-(m-\sigma^2))^2}{2\sigma^2}}. \tag{3.24}$$

The claim follows directly from the trivial identities

$$\begin{aligned}(s+r)^y s &= (s+r)^{y+1} - r(s+r)^y,\\ (s+r)^y s^2 &= (s+r)^{y+2} - 2r(s+r)^{y+1} + r^2(s+r)^y.\end{aligned}$$

The desired identities follow from the definitions and the recursions in (3.23) by

$$\begin{aligned}\bar{s} &= \frac{J_{y,1}}{J_{y,0}} = \frac{\alpha(I_{y+1} - rI_y)}{\alpha I_y} = \frac{I_{y+1}}{I_y} - r,\\ C_s &= \frac{J_{y,2}}{J_{y,0}} - \bar{s}^2 = \frac{\alpha(I_{y+2} - 2rI_{y+1} + r^2 I_y)}{\alpha I_y} - \Big(\frac{I_{y+1}}{I_y} - r\Big)^2 = \frac{I_{y+2}}{I_y} - \Big(\frac{I_{y+1}}{I_y}\Big)^2.\end{aligned}$$

This completes the proof. □

However, directly evaluating I_y can still be numerically unstable for large y. To avoid the potential instability, we develop a stable recursive scheme in Lemma 3.4.1.

Lemma 3.4.1. *For $y \geq 2$, the following recursion formula holds*

$$I_y = (m - \sigma^2 + r)I_{y-1} + \sigma^2(y-1)I_{y-2} + \frac{\sigma^2(b+r)^{y-1}}{\sqrt{2\pi\sigma^2}} e^{-\frac{(b-m+\sigma^2)^2}{2\sigma^2}}. \tag{3.25}$$

Proof. Let $c = m - \sigma^2$, $d = \sigma^2$ and $f(s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(s-c)^2}{2d}}$. The definition of I_y implies

$$\begin{aligned}I_y &= \int_b^\infty (s+r)^y f(s)\mathrm{d}s = \int_b^\infty (s+r)^{y-1}(d\frac{s-c}{d} + c + r) f(s)\mathrm{d}s\\ &= -d\int_b^\infty (s+r)^{y-1}(-\frac{s-c}{d}) f(s)\mathrm{d}s + (c+r)\int_b^\infty (s+r)^{y-1} f(s)\mathrm{d}s.\end{aligned} \tag{3.26}$$

Next we employ the trivial identity $\frac{\mathrm{d}}{\mathrm{d}s} f(s) = -\frac{s-c}{d} f(s)$ and apply integration by parts to the first

term

$$\begin{aligned}
&\int_b^\infty (s+r)^{y-1}(-\frac{s-c}{d})f(s)\mathrm{d}s \\
&= (s+r)^{y-1}f(s)|_b^\infty - \int_b^\infty (y-1)(s+r)^{y-2}f(s)\mathrm{d}s \\
&= -(b+r)^{y-1}f(b) - (y-1)I_{y-2}.
\end{aligned} \tag{3.27}$$

Collecting the terms shows the desired recursion on the integral I_y. □

Remark 3.4.1. *For $b = -r$, we have a simplified recursive formula for I_y:*

$$I_y = (m - \sigma^2 + r)I_{y-1} + \sigma^2(y-1)I_{y-2}.$$

Lemma 3.4.1 uses a two-term linear recurrence relation for I_y's. The coefficients of I_{y-1} and I_{y-2} are raised by power when expanding I_y in terms of I_0 and I_1, and thus the computation of I_y is susceptible to the evaluation errors of I_0 and I_1 for large coefficients. This motivates a reciprocal recursive scheme for $r = 0$ or $b = -r$ by introducing a ratio sequence $\{L_y\}_y$ defined by

$$L_y = \frac{yI_{y-1}}{I_y}. \tag{3.28}$$

Note that L_y also admits a recursive scheme

$$L_y = \frac{y}{(m - \sigma^2 + r) + \sigma^2 L_{y-1}}, \tag{3.29}$$

and further I_y can be recovered from $\{L_y\}$ by

$$\ln I_y = \ln y! + \ln I_0 - \sum_{i=1}^{y} L_i. \tag{3.30}$$

We can compute $\bar{s}$ and C_s directly from L_y. The identities follow from straightforward computation.

Theorem 3.4.2. *If $r = 0$ or $b = -r$, the ratios for calculating $\bar{s}$ and C_s are given by*

$$\frac{I_{y+1}}{I_y} = (m - \sigma^2 + r) + \sigma^2 L_y \quad \textit{and} \quad \frac{I_{y+2}}{I_y} = e^{\ln(y+1)+\ln(y+2)-\ln L_{y+1}-\ln L_{y+2}}. \tag{3.31}$$

Last, we discuss stable methods for computing the first three integrals I_0, I_1 and I_2, which are needed for the recursion. We consider three different forms and use them separately according to the integration range with respect to the auxiliary variable

$$\eta = \frac{\sigma^2 - m + b}{\sqrt{2\sigma^2}}.$$

The formulae are listed in Table 3.1, where erf and erfc denote the error function and complementary error function, respectively, and $\mathrm{erfcx}(\eta) = e^{\eta^2}(1-\mathrm{erf}(\eta))$. Since the value of $1-\mathrm{erf}(\eta)$ is vanishingly small for large η value, we use Scheme 2 to avoid underflow. Scheme 3 is useful when the η value is large, since both $1-\mathrm{erf}(\eta)$ and $\mathrm{erfc}(\eta)$ suffer from numerical underflow. Note that when η is small, Scheme 3 is not as accurate as Scheme 2, so we use Scheme 2 in the intermediate range. In our experiments, we use Scheme 1 for $\eta \in (-\infty, 5)$, Scheme 2 for $\eta \in [5, 26)$ and Scheme 3 for $\eta \in (26, \infty)$. To use Scheme 3, we construct $\tilde{I}_i = \frac{I_i}{I_0}$, $i = 0, 1, 2$, and $\tilde{L}_y = \frac{y\tilde{I}_{y-1}}{\tilde{I}_y}$, $y \in \mathbb{N}_+$. Then similar identities for computing $\bar{s}$ and C_s hold, i.e.,

$$\bar{s} = \frac{\tilde{I}_{y+1}}{\tilde{I}_y} - r \quad \text{and} \quad C_s = \frac{\tilde{I}_{y+2}}{\tilde{I}_y} - \Big(\frac{\tilde{I}_{y+1}}{\tilde{I}_y}\Big)^2, \tag{3.32}$$

with $\frac{\tilde{I}_{y+1}}{\tilde{I}_y} = (m - \sigma^2 + r) + \sigma^2 \tilde{L}_y$ and $\frac{\tilde{I}_{y+2}}{\tilde{I}_y} = e^{\ln(y+1)+\ln(y+2)-\ln\tilde{L}_{y+1}-\ln\tilde{L}_{y+2}}$.

Table 3.1: Three schemes for evaluating I_0, I_1 and I_2.

scheme	formulae	condition
1	$I_0 = \frac{1}{2}(1-\mathrm{erf}(\eta)), \qquad I_1 = \sqrt{\frac{\sigma^2}{2\pi}} e^{-\eta^2} + \frac{m-\sigma^2+r}{2}(1-\mathrm{erf}(\eta))$ $I_2 = \sqrt{\frac{\sigma^2}{2\pi}}(m-\sigma^2+b+2r)e^{-\eta^2} + \frac{(m-\sigma^2+r)^2+\sigma^2}{2}(1-\mathrm{erf}(\eta))$	$\eta < 5$
2	$I_0 = \frac{1}{2}\mathrm{erfc}(\eta), \qquad I_1 = \sqrt{\frac{\sigma^2}{2\pi}} e^{-\eta^2} + \frac{m-\sigma^2+r}{2}\mathrm{erfc}(\eta)$ $I_2 = \sqrt{\frac{\sigma^2}{2\pi}}(m-\sigma^2+b+2r)e^{-\eta^2} + \frac{(m-\sigma^2+r)^2+\sigma^2}{2}\mathrm{erfc}(\eta)$	$5 \le \eta \le 26$
3	$\tilde{I}_0 = 1, \qquad \tilde{I}_1 = \sqrt{\frac{2\sigma^2}{\pi}} \frac{1}{\mathrm{erfcx}(\eta)} + m - \sigma^2 + r$ $\tilde{I}_2 = \sqrt{\frac{2\sigma^2}{\pi}} \frac{m-\sigma^2+b+2r}{\mathrm{erfcx}(\eta)} + (m-\sigma^2+r)^2 + \sigma^2$	$\eta > 26$

3.4.2 Laplace Potential

Since Laplace distributions do not have the factor s^y, the evaluation of related integrals does not entail a recursive scheme. Below we employ the idea in [124, 52] and derive the formulae for evaluating the 1D integrals for the Laplace potential $t(x) = \frac{\mu}{2} e^{-\mu|\ell^t x|}$.

For any fixed $\ell \in \mathbb{R}^n$, we divide the whole space $\mathbb{R}^n$ into two disjoint half-spaces V_+ and V_-, i.e.,

$$\mathbb{R}^n = V_+ \cup V_-, \quad \text{with } V_+ = \{x | \ell^t x > 0\} \text{ and } V_- = \{x | \ell^t x \le 0\}. \tag{3.33}$$

Then we split the Laplace potential $t(x)$ into

$$t(x) = \frac{\mu}{2} e^{-\mu \ell^t x} \mathbf{1}_{V_+}(x) + \frac{\mu}{2} e^{\mu \ell^t x} \mathbf{1}_{V_-}(x). \tag{3.34}$$

The integrals involving $t(x)\mathcal{N}(s|m,\sigma^2)$ can be divided into two parts:

$$\begin{aligned}\int_{\mathbb{R}_+}\frac{\mu}{2}s^i e^{-\mu s}\mathcal{N}(s|m,\sigma^2)\mathrm{d}s &= \frac{\mu}{2}e^{\frac{\mu^2\sigma^2}{2}}\underbrace{e^{-\mu m}\int_{\mathbb{R}_+}s^i\mathcal{N}(s|m-\mu\sigma^2,\sigma^2)\mathrm{d}s}_{:=I_i^+},\\ \int_{\mathbb{R}_-}\frac{\mu}{2}s^i e^{\mu s}\mathcal{N}(s|m,\sigma^2)\mathrm{d}s &= \frac{\mu}{2}e^{\frac{\mu^2\sigma^2}{2}}\underbrace{e^{\mu m}\int_{\mathbb{R}_-}s^i\mathcal{N}(s|m+\mu\sigma^2,\sigma^2)\mathrm{d}s}_{:=I_i^-}.\end{aligned} \tag{3.35}$$

By the change of variable $t=\frac{s-m\pm\mu\sigma^2}{\sigma}$ for $I_i^\pm$ respectively, we have

$$I_i^+ = \frac{e^{-\mu m}}{\sqrt{2\pi}}\int_{-\frac{m}{\sigma}+\mu\sigma}^{+\infty}(\sigma t+m-\mu\sigma^2)^i e^{-\frac{t^2}{2}}\mathrm{d}t, \tag{3.36}$$

$$I_i^- = \frac{(-1)^i e^{\mu m}}{\sqrt{2\pi}}\int_{\frac{m}{\sigma}+\mu\sigma}^{+\infty}(\sigma t-m-\mu\sigma^2)^i e^{-\frac{t^2}{2}}\mathrm{d}t. \tag{3.37}$$

The integrals can be expressed using the cumulative distribution function Φ of the standard Gaussian distribution. We shall view $I_i^\pm$ as functions of m and let $I_i = I_i^+(m)+(-1)^i I_i^+(-m)$. Then we have

$$\bar{s}=\frac{I_1}{I_0}\quad\text{and}\quad C_s=\frac{I_2}{I_0}-\left(\frac{I_1}{I_0}\right)^2. \tag{3.38}$$

To avoid the potential underflow of direct evaluation of Φ, we use the following well known (divergent) asymptotic expansion [2, item 7.1.23]

$$\begin{aligned}1-\Phi(\eta) &= \int_\eta^\infty e^{-\frac{t^2}{2}}\mathrm{d}t = e^{-\frac{\eta^2}{2}}\left(\eta^{-1}+\sum_{k=1}^{\infty}\frac{(-1)^k(2k-1)!}{2^k(k-1)!}\eta^{-(2k+1)}\right)\\ &= \mathcal{N}(\eta|0,1)\eta^{-1}\underbrace{\sum_{n=0}^{\infty}(-1)^n(2n-1)!!\eta^{-2n}}_{:=g(\eta)}.\end{aligned}$$

This formula follows by integration by parts, and allows accurate evaluation for large positive η. It was shown in [51] that the error of evaluating $1-\Phi(\eta)$ with a truncation of the asymptotic expansion is less than 10^{-11} for $\eta>5$ with more than 8 terms in the summation of $g(\eta)$. For $\eta\le 5$, $1-\Phi(\eta)$ can be accurately evaluated directly. Then we introduce a ratio

$$\alpha=\frac{I_0^+(-|m|)}{I_0^+(|m|)}=e^{2\mu|m|}\frac{(\mu\sigma^2-|m|)g(\mu\sigma+\frac{|m|}{\sigma})}{(\mu\sigma^2+|m|)g(\mu\sigma-\frac{|m|}{\sigma})}.$$

With the ratio α, the two fractions $\frac{I_1}{I_0}$ and $\frac{I_2}{I_0}$ can be evaluated by

$$\begin{aligned}\frac{I_1}{I_0} &= m+\mu\sigma^2\mathrm{sgn}(m)\Big(1-\frac{2}{1+\alpha}\Big),\\ \frac{I_2}{I_0} &= -2\mu\sigma^3\frac{1}{\sqrt{2\pi}}e^{-\frac{m^2}{2\sigma^2}}e^{-\frac{\mu^2\sigma^2}{2}}I_0^{-1}+(\sigma^2+\mu^2\sigma^4-m^2)+2m\frac{I_1}{I_0}.\end{aligned}$$

To avoid potential numerical instability of the first term in $\frac{I_2}{I_0}$, we use the identity

$$-\frac{2\mu\sigma^3}{\sqrt{2\pi}}e^{-(\frac{m^2}{2\sigma^2}+\frac{\mu^2\sigma^2}{2})}I_0^{-1}=\frac{-2\mu\sigma^2(-|m|+\mu\sigma^2)}{g(-\frac{|m|}{\sigma}+\mu\sigma)(1+\alpha)}.$$

To avoid potential numerical instability of the term $\sigma^2+\mu^2\sigma^4$, we use the exp-log trick

$$\sigma^2+\mu^2\sigma^4=\exp\Big(-2\log\frac{1}{\mu\sigma^2}+\log\big(1+\frac{1}{\mu^2\sigma^2}\big)\Big),$$

where $\log(1+\frac{1}{\mu^2\sigma^2})$ is evaluated by the `MATLAB` builtin function `log1p`. Thus, $\bar{s}$ and C_s can be evaluated by

$$\bar{s}=\frac{I_1}{I_0}\quad\text{and}\quad C_s=-\frac{2\mu\sigma^2(-|m|+\mu\sigma^2)}{g(-\frac{|m|}{\sigma}+\mu\sigma)(1+\alpha)}+\exp[-2\log\frac{1}{\mu\sigma^2}+\log(1+\frac{1}{\mu^2\sigma^2})]-(m-\frac{I_1}{I_0})^2.$$

3.5 Numerical Experiments

Now we present numerical results to illustrate the features, i.e., convergence, accuracy and feasibility for large-scale problems, of the EP algorithm for constrained Poisson distribution. For the first two cases, we adopt the following classical one-dimensional example [108]: a Fredholm integral equation of the first kind with the kernel $K(s,t)=\phi(s-t)$ with

$$\phi(s)=\begin{cases}10+10\cos\frac{\pi}{3}s, & |s|<3,\\ 0, & |s|\geq 3,\end{cases}$$

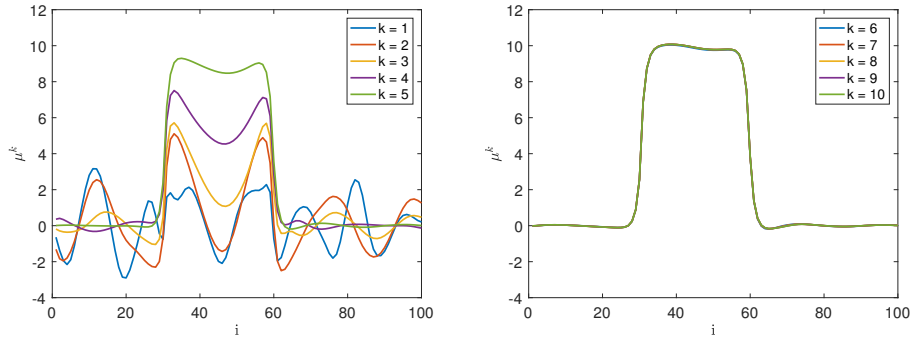
and the integration interval is $[-6,6]$. The exact solution $x(t)$ is given by $\phi(t)$. The problem is discretised by a standard piecewise constant Galerkin method, and the resulting problem is size 100, i.e., $x\in\mathbb{R}^{100}$ and $A\in\mathbb{R}^{100\times 100}$. It is mildly ill-posed, and has been used as a benchmark problem in several studies. In the numerical implementation, we employ the parameterisation with natural parameters, i.e., Algorithm 3, which appears to be numerically more robust. The experiments are conducted on a desktop with Intel i7-7700K CPU 4.20GHz×8.

3.5.1 Convergence of EP Algorithm

First, we examine the convergence of EP algorithm. The convergence of EP algorithms is a long outstanding theoretical issue, and remains largely elusive. Thus, we present an experimental evaluation of the convergence behaviour. Recall that EP algorithm has two level of iterations, i.e., outer and and inner, and in each outer iteration, EP visits all factors once. We denote the mean and covariance after the kth outer iteration by μ^k and C^k, and monitor their convergence.

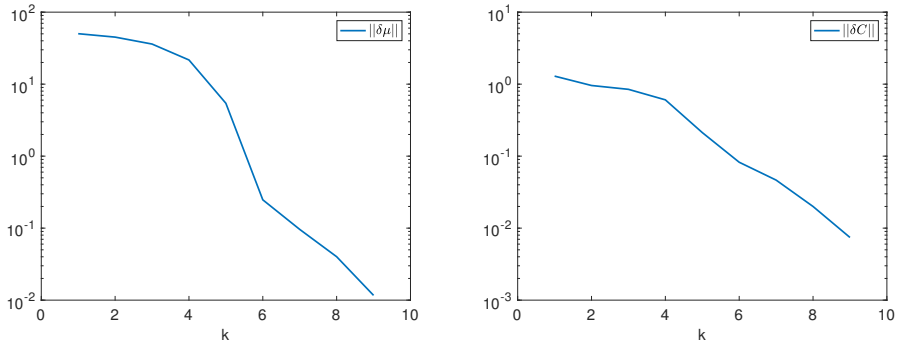
In Figs. 3.1 and 3.2, we show the numerical results for EP convergence for the first ten outer iterations, each corresponding to one loop over all data points. It is observed from Fig. 3.1 that the mean μ^k converges fairly rapidly during the first few iterations, capturing the essential features

of the mean, and it reaches convergence after five iterations, since the recovered mean is hardly distinguishable afterwards, cf. Fig. 3.1(b). The plots are further confirmed by the errors of the iterate relative to the converged iterate (μ^*, C^*). The errors $\delta\mu = \mu^k - \mu^*$ and $\delta C = C^k - C^*$ are measured by the L^2-norm and spectral norm, respectively. Hence, both the mean and covariance converge rapidly, showing the fast convergence of EP.



(a) μ^1 to μ^5 (b) μ^6 to μ^{10}

Figure 3.1: The convergence of the mean μ^k by EP after k outer iterations.



(a) mean μ (b) covariance C

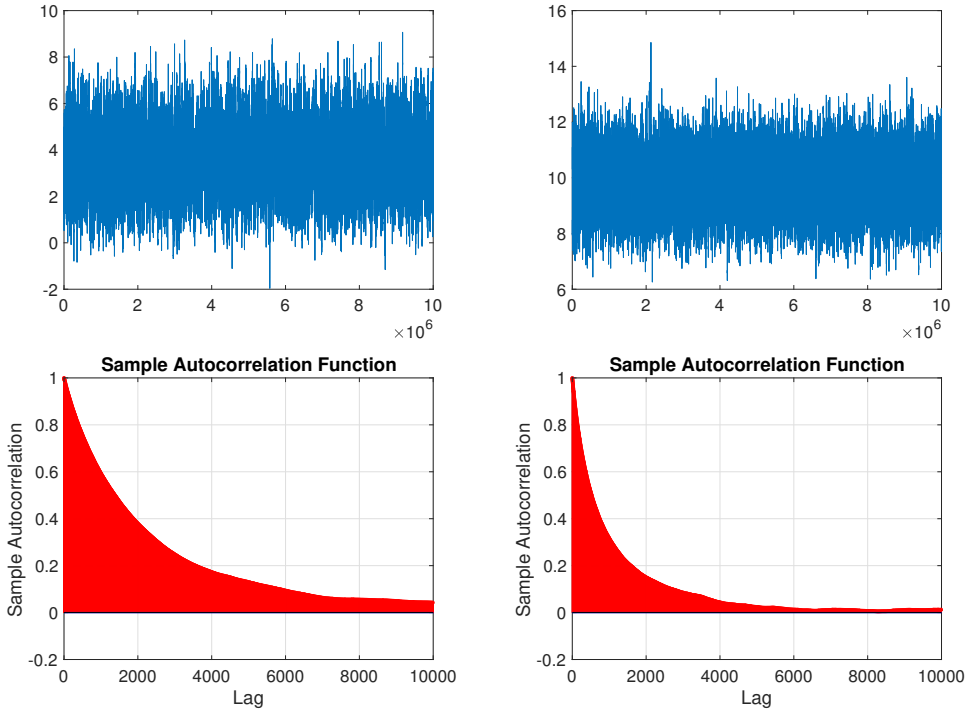
Figure 3.2: The convergence of the mean μ and covariance C after each outer iteration.

3.5.2 Comparison Between EP and the True Posterior

Numerically, the accuracy of EP has found to be excellent in several studies [112, 52], although there is still no rigorous quantification of the error. We provide an experimental evaluation of its accuracy by comparing the EP results with the true posterior distribution. The features of the true posterior distribution $p(x|y)$ are captured by Markov chain Monte Carlo (MCMC) [90, 119], which is known to be asymptotically exact, and the MAP estimation. To obtain MCMC results, we run a random walk Metropolis-Hastings sampler with the Gaussian steps. The step size in the Metropolis-Hastings algorithm is optimised so that the acceptance ratio is close to 0.23 in order to ensure good convergence of the MCMC chain. The MCMC chain is run for a length of 2×10^7, and the last 10^7 samples are used for computing the summarising statistics, e.g., mean and covariance. From Fig.

3.3, one can observe that there is no periodic patterns in the trace plots, which supports convergence of the chain. However, the autocorrelations of the sample trajectory decay not very fast, which might be the result of the improper prior.

To compare the Gaussian approximation by EP and MCMC results, we present the mean, MAP, covariance and marginal 95% posterior credible intervals. It is observed that both approximations concentrate in the same region, and the shape and magnitude of the posterior credible intervals / covariance are mostly comparable with each other, cf. Figs. 3.4 and 3.5. However, there are noticeable differences in the mean approximation: the mean by EP is fairly close to being piecewise constant, which differs from that by MCMC. Theoretically, the MAP estimate with a TV penalty tends to be piecewise constant, but the posterior mean is not necessarily so. So EP provides an intermediate approximation between the MAP and posterior mean. In comparison with the MAP, EP provides not only a point estimate, but also the associated uncertainty using the covariance. Further, the covariance is clearly diagonal dominant, which suggests the use of a banded covariance or its Cholesky factor for potentially speeding up the algorithm. The magnitudes of the diagonals are also largely comparable, even though that by EP are slightly smaller.



(a) the 30-th element (b) the 50-th element

Figure 3.3: Trace plots and autocorrelation of MCMC samples the 30-th and 50-th element.

3.5.3 Medium Size Test

Now we illustrate the feasibility of the approach on larger scale problems. We consider images of size 128×128 pixels, i.e., $x \in \mathbb{R}^{16384}$. The ground-truth images are the `Shepp-Logan`

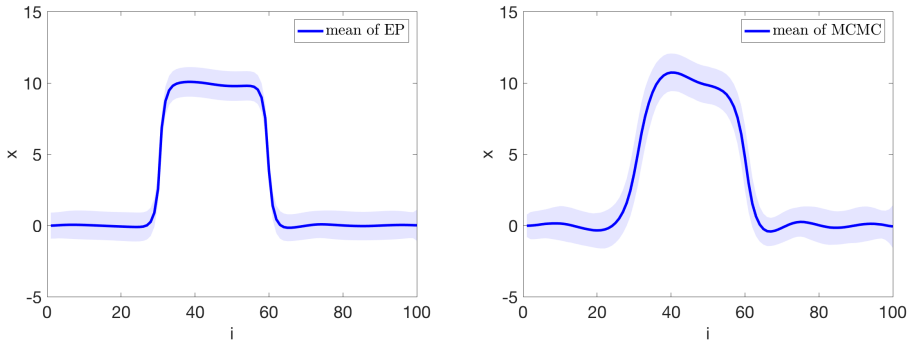
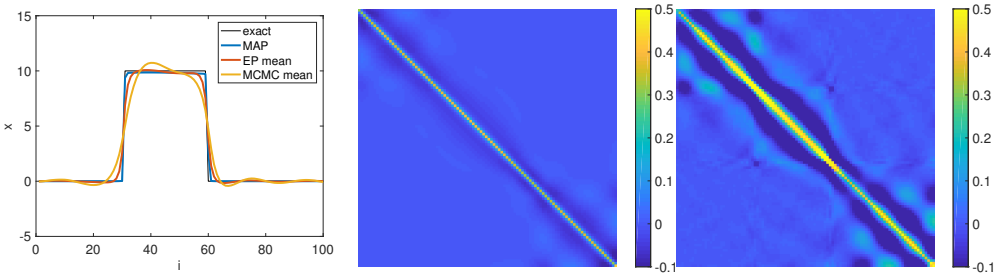


Figure 3.4: Comparisons of mean and 0.95 posterior credible intervals between EP and MCMC for `Phillips` test



(a) EP and MCMC means (b) EP covariance (c) MCMC covariance

Figure 3.5: Comparisons of mean and covariance of EP and MCMC for `Phillips` test

phantom, one `PET` phantom taken from [40] and `IRT` phantom from the `Michigan Image Reconstruction Toolbox`[1]. The forward map A is given by discrete Radon transforms, and is generated using `MATLAB` built-in function `radon` with 185 projections per angle and three different angle settings, i.e. $[0:2:179]$, $[0:4:179]$ and $[0:8:179]$, and the corresponding matrix is of size $A \in \mathbb{R}^{16650\times 16384}$, $A \in \mathbb{R}^{8325\times 16384}$ and $A \in \mathbb{R}^{4255\times 16384}$. The original image, the sinogram (i.e., the image after Radon transform) and the observed Poisson data are shown in Figs. 3.6, 3.8 and 3.10.

We present reconstructions for the MAP of the posterior distribution $p(x|y)$, where the hyperparameter μ is taken to be the same for both approaches in order to ensure a fair comparison. The MAP reconstructions are computed by a limited-memory BFGS algorithm. We measure the accuracy of the reconstruction x^* relative the ground truth $x^\dagger$ by the standard L^2-error $||x^* - x^\dagger||_2$, the structural similarity (SSIM) index (by `MATLAB` built-in `ssim`), and peak signal-to-noise ratio (PSNR) (by `MATLAB` built-in `psnr` with peak value 1). The numerical results are summarised in Tables 3.2, 3.3 and 3.4. The experiments show clearly the feasibility of the proposed approach for handling medium size images.

It is observed that the estimated mean by the EP is mostly comparable with the MAP estimate in all three metrics. For both approaches, the reconstruction quality improves steadily as the number of projection angle increases, for which the data becomes more informative.

[1] `https://web.eecs.umich.edu/~fessler/code/`, last accessed on July 30, 2018.

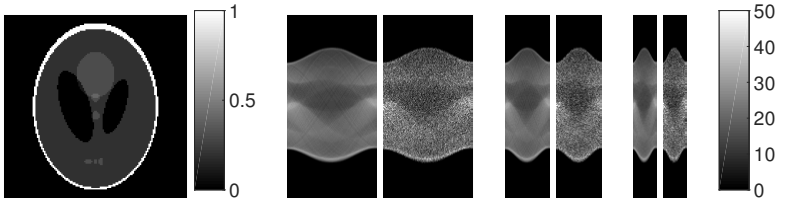


Figure 3.6: The exact image, sinograms and observed data with three different A's for `Shepp-Logan` phantom.

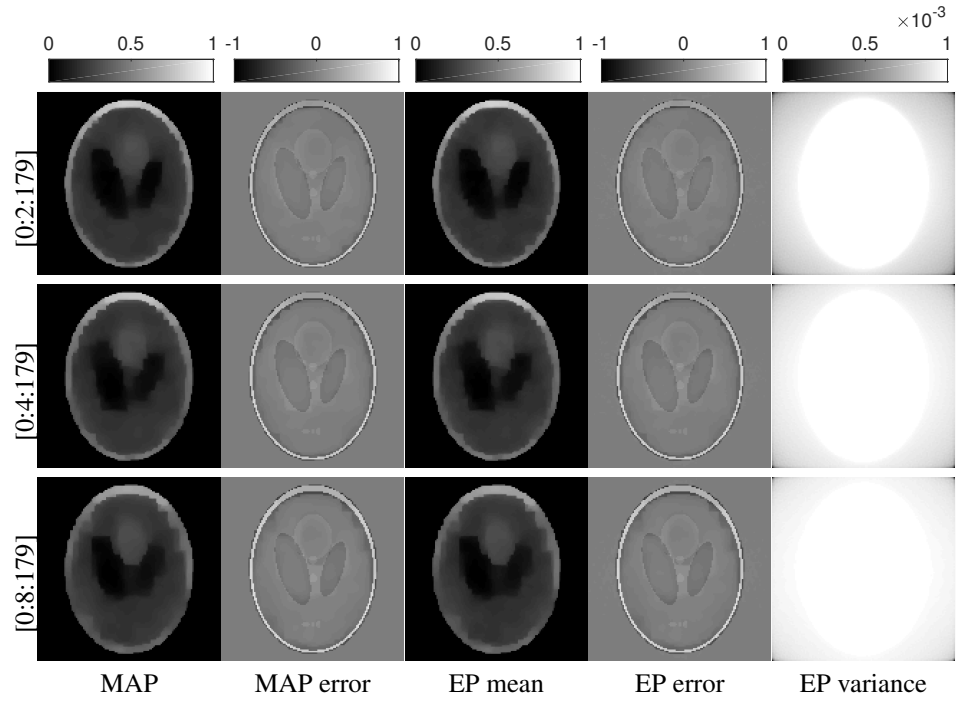


Figure 3.7: MAP vs EP with anisotropic TV prior for the `Shepp-Logan` phantom.

3.6 Conclusion

In this work, we have developed an approximate inference procedure for the constrained Poisson likelihood which arises in emission tomography. The approach is based on expectation propagation developed in the machine learning community. The detailed derivation of the algorithms, complexity and their stable implementation are given, for the case of a Laplace type prior. The approach is illustrated with numerical experiments, which show that it converges rapidly and can deliver results

Table 3.2: Comparisons between EP mean and MAP for the `Shepp-Logan` phantom.

angle	[0:2:179]		[0:4:179]		[0:8:179]	
μ	6e0		4e0		3e0	
Method	EP	MAP	EP	MAP	EP	MAP
L2 Error	5.32	5.36	5.64	5.67	6.09	6.11
SSIM	0.74	0.78	0.70	0.75	0.67	0.72
PSNR	18.58	18.53	17.97	17.93	17.29	17.27
CPU time (s)	80187.88	124.44	46031.95	55.55	29274.16	27.23

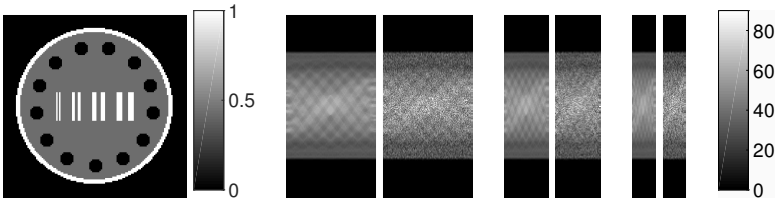


Figure 3.8: The exact image, sinograms and observed data with three different A's for the `PET` phantom.

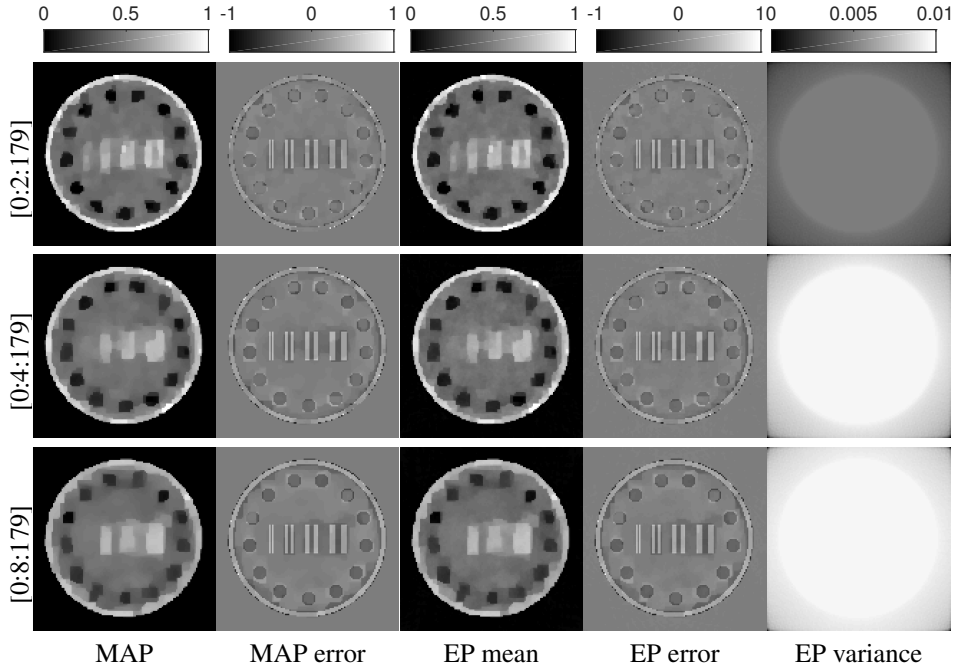


Figure 3.9: MAP vs EP with anisotropic TV prior for the `PET` phantom.

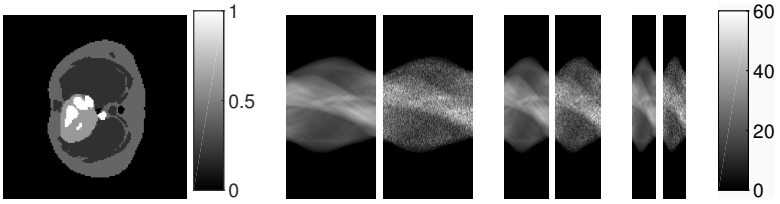


Figure 3.10: The exact image, sinograms and observed data with three different A's for `IRT` phantom.

Table 3.3: The comparisons between EP mean and MAP for the `PET` phantom

angle	[0:2:179]		[0:4:179]		[0:8:179]	
μ	1.6e0		1.4e0		1.2e0	
Method	EP	MAP	EP	MAP	EP	MAP
L2 Error	7.37	7.45	8.55	8.64	8.81	8.87
SSIM	0.72	0.81	0.61	0.75	0.57	0.70
PSNR	19.82	19.79	18.42	18.35	17.35	17.28
CPU time (s)	91263.00	110.05	53863.77	78.69	31537.05	28.20

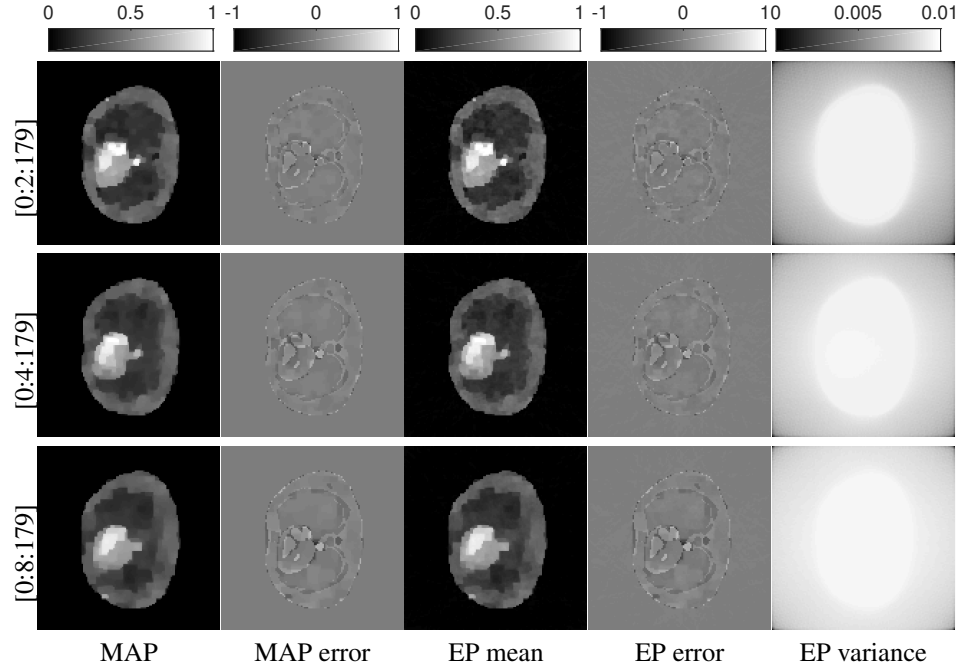


Figure 3.11: MAP vs EP with anisotropic TV prior for the `IRT` phantom.

Table 3.4: The comparisons between EP mean and MAP for the `IRT` phantom.

angle	[0:2:179]		[0:4:179]		[0:8:179]	
μ	1.6e0		1.4e0		1.2e0	
Method	EP	MAP	EP	MAP	EP	MAP
L2 Error	2.07	2.07	2.29	2.30	2.89	2.91
SSIM	0.65	0.88	0.65	0.87	0.63	0.84
PSNR	25.87	26.20	25.44	25.62	24.26	24.36
CPU time (s)	82017.89	101.87	57209.36	51.00	29224.76	23.14

comparable with MAP and MCMC, and it can handle medium size images.

There are several avenues for future works. First, it is of enormous interest to analyse the convergence rate and accuracy of EP, and more general approximate inference techniques, e.g., variational Bayes, which have all achieved great practical successes but largely defied theoretical analysis. Second, it is important to further extend the flexibility of EP algorithms to more complex posterior distributions, e.g., lack of projection form. One notable example is isotropic total variation prior that appears frequently in practical imaging algorithms. This may require introducing an additional layer of approximation in the spirit of iteratively reweighted least-squares or Monte Carlo computation of the low-dimensional integrals. Third, many experimental studies show that EP converges very fast, with convergence reached within five outer iterations for the Poisson model under considerations. However, the overall $O(mn^2)$ computational complexity is still very high for all current implementations [53], and not directly scalable to really big images. Hence, it is of great interest to accelerate the algorithms by identifying suitable structures on the problem, e.g., the intrinsic low-rank structure of the forward map A and the diagonal dominance of the posterior covariance. Moreover, the

stability of 1 dimensional integration schemes are developed based on empirical observation in the sense that they could convergence while traditional quadrature rules could not. It is of great interest to deepen the understanding of and analyse stability.

Chapter 4

Probabilistic Iterative Networks for Inverse Problems

4.1 Introduction

Machine learning techniques, predominantly deep neural networks (DNNs), have received much attention for solving inverse problems in recent years, and delivered state-of-the-art reconstruction performance on many classical inverse problems, including image denoising [142, 143], image deblurring [138, 122], super-resolution [37], and challenging practical inverse problems, e.g., low-dose / sparse-data computed tomography [74, 28] and magnetic resonance imaging [66]. See the recent surveys [96, 93, 13] and the long lists of references therein for various important advances on DNNs for inverse problems and successful practical applications. The most prominent ideas underpinning these exciting developments include end-to-end processing and unrolling iterative algorithms (gradient descent, proximal gradient, primal-dual gradient descent, ADMM etc.). Most existing works on DNN-based inversion aim at providing one point estimate of the unobservable (i.e., the signal / image of interest) for a given observation, by viewing DNNs as deterministic mappings, by exploiting their extraordinary approximation capability (as well as rich training data). As discussed in Section 1.1, a fully probabilistic treatment for uncertainty quantification (UQ) to assess the reliability of one specific inverse solution is necessary. This important piece of uncertainty information is not directly available from most existing DNN-based inversion methods.

The Bayesian framework provides a systematic yet very flexible framework for the challenging UQ task, and has been the method of choice for UQ of inverse problems [73, 128]. The conventional Bayesian setting often involves explicit likelihood and prior constructions. Recent advances in Bayesian inference leverage powerful deep generative modelling tools, e.g., Wasserstein generative adversarial networks (GANs) [12, 4] and variational auto-encoders (VAEs) [80, 126, 141], and admit implicit priors and noise models. Distinctly, within these frameworks, once the network is trained (often in an expensive offline stage), generating one sample of the unobservable reduces to one DNN feedforward propagation, which is computationally very light and for multiple samples, it

can be run in parallel. Thus, these techniques hold enormous potentials for UQ of inverse problems, and it is highly desirable to develop analogues for DNN-based techniques. However, the rigorous UQ of data-driven inversion techniques within a Bayesian framework is still very challenging.

The first challenge stems from high complexity of posterior distributions involving DNNs. Conventionally, in Bayesian inversion, both likelihood and prior are given explicitly (or hierarchically). The likelihood is derived from the statistical model of the forward observation process, assuming that the noise statistics and underlying physical principles of the imaging modality are well calibrated. Nonetheless, deriving precise likelihoods can be nontrivial, e.g., due to complex corruption process. Meanwhile, how to stipulate statistically meaningful yet explicit priors is a long-standing open problem. Samples from commonly adopted priors, e.g. Gaussian, Laplace and total variation, actually do not resemble natural images at all. Learning based approaches instead model implicitly these knowledge in a data-driven way from the training data, and thus, the resulting posterior is intrinsically implicit.

The second challenge is related to the physical laws. One distinct feature of many inverse problems is the presence of forward maps, when compared with tasks in machine learning. The forward maps often describe fundamental physical laws, and serve as a part of established prior knowledge. It is also implicitly encoded in the pair of ground-truth data and observations, and DNNs can extract it but only in a black-box way, only if given a large volume of training data. Thus it is important to directly inform DNNs with the physical laws, which may enable using a small amount of training data.

In this work, we develop a novel computational framework, termed as Probabilistic Iterative Networks (PIN), for UQ of inverse problems. It employs the conditional variational auto-encoder (CVAE) loss [126] to gracefully address the challenges. Minimising the CVAE loss is equivalent to minimising an upper bound of the expected reversed Kullback-Leibler divergence [141]. This interpretation indicates that the obtained samples are drawn from an approximate posterior distribution. Specifically, the network aggregates the information in the forward map, observation and samples of a low-dimensional random variable that is conditionally dependent on the observation, and recurrently refines the (stochastic) reconstructions using the information in the input. The framework has several distinct features: (i) The low-dimensionality of the probabilistic encoder ensures good scalability of the framework; (ii) The network is flexible in the sense that it allows training on small-scale datasets while inference on large-scale problems; (iii) The trained network serves as an efficient sampler from an approximate posterior distribution in the sense of variational inference. In summary, the framework incorporates forward map(s), handles implicit model constructions, and is very flexible with training datasets. Thus, it is applicable to a wide range of practical inverse problems. In Section 4.4, we illustrate these desirable features with experiments on one established medical imaging modality – positron emission tomography (PET), and confirm that the generated samples are of very high quality in terms of point estimation and uncertainty quantification, when

compared with several state-of-the-art benchmarks. To the best of our knowledge, this is the first flexible DNN-based framework for UQ of general inverse problems.

Now we situate this work in the context of quantifying uncertainty for deep learning-based inversion techniques. As discussed in Section 1.1, while different kinds of uncertainty are studied in the literature, we focus on aleatoric uncertainty in this study. Thus, it differs from many existing Bayesian treatments for neural networks, e.g. Bayesian neural networks (BNNs) for *epistemic* uncertainty [55, 21, 48] and prior networks for out-of-distribution inputs [95]. BNNs model the uncertainty on network weights, and preserve the benefits of Bayesian principles but often at the expense of compromising performance [104]. To rigorously justify the distribution modelled by DNNs for *aleatoric* uncertainty, proper Bayesian interpretations of loss functions used in training DNNs (in connection with the target posterior) are needed. Recently, Adler and Öktem [4] proposed a loss function whose minimisation is equivalent to minimising the Wasserstein 1-distance between the posterior distribution and the approximation under Lipschitz conditions on DNNs, which, however, are not easy to enforce [57]. In contrast, the CVAE loss used in this work does not impose any restrictions on DNNs.

The rest of the chapter is organised as follows. In Section 4.2, we describe the problem setting and notations. Then in Section 4.3, we review auto-encoding variational Bayes and develop the new approach, i.e., Probability Iterative Networks. In Section 4.4, we showcase the framework with numerical experiments and in Section 4.5, we conclude the chapter with additional discussions.

4.2 Problem Formulation and Notations

Now we set the stage of this work, i.e., finite-dimensional inverse problems. Let $x \in \mathbb{R}^n$ be the unobservable of interest, and $y \in \mathbb{R}^m$ be the observable (data). Generally, the dimensions of x and y are different, and vary with the discretisations of the underlying continuous model. The forward operator $\mathscr{A} : \mathbb{R}^n \to \mathbb{R}^m$ linearly maps the unobservable x to an observable y. Its adjoint $\mathscr{A}^* : \mathbb{R}^m \to \mathbb{R}^n$ maps an element in $\mathbb{R}^m$ back to the unobservable space $\mathbb{R}^n$. For example, in computed tomography (CT), the forward map $\mathscr{A}$ is given by the Radon transform and the adjoint is a dual Radon transform (a.k.a. backprojection), whereas in electrical impedance tomography, both forward map and adjoint are described implicitly by differential equations.

In practice, the observation y is a noisy version of the exact data $\mathscr{A}(x)$:

$$y = \eta(\mathscr{A}(x)),$$

where $\eta(\cdot)$ denotes a general corruption process by (possibly unknown type) noise, e.g., Gaussian, Poisson, and Salt and Pepper noise, or the mixtures thereof. In probability, the corruption process $\eta(\cdot)$ is encoded by a conditional distribution $p^*(y|x)$. Here the unobservable x is a random variable, with its prior distribution $p^*(x)$. In the proposed framework, we only require

(i) we can draw samples from the joint distribution $p^*(x,y) := p^*(y|x)p^*(x)$;

(ii) we can access the forward map $\mathscr{A}(\cdot)$ and its adjoint $\mathscr{A}^*(\cdot)$ for each data pair (x,y)

This setting covers a general family of likelihood and prior and generalises the problem settings in Chapter 2 and Chapter 3. In many inverse problems, the forward maps are accurately described by established physical theories but the corruption process is not well studied. The setting is especially suitable for situations where measurements can be acquired but the corruption process is challenging / inconvenient to calibrate, and instead, inverse models can learn the process from physically derived data.

To generate training samples from the joint distribution $p^*(x,y)$, one can collect measurements for samples of the unobservable x drawn from the prior $p^*(x)$ (i.e., physically derived data), without explicitly knowing the corruption process $\eta(\cdot)$. Also it can use simulated data with a known noise model $\eta(\cdot)$ using samples from $p^*(x)$. Thus, in the training data tuple $\{(x_i, y_i, \mathscr{A}_i, \mathscr{A}_i^*)\}_{i=1}^N$, y_i is generated by $\mathscr{A}_i(x_i)$ (together with η) and $\mathscr{A}_i^*$ is the adjoint of $\mathscr{A}_i$. The presence of operators $\mathscr{A}$ and $\mathscr{A}^*$ is an important difference from standard supervised learning in machine learning.

When training DDNs, we input the operators $(\mathscr{A}_i, \mathscr{A}_i^*)$ as partial model knowledge to avoid overfitting the training data. In practical inverse problems, e.g. medical imaging, labeled training data are often expensive to acquire in a large volume, if not impossible at all. Thus, it is highly desirable that the learning based methods can be trained on small datasets and can generalise the learned knowledge to unseen data. By explicitly building the forward operators into networks, DNNs directly respect the underlying physical principles, thereby reducing the requisite amount of training data.

Below we use $h(\cdot)$ to denote a DNN, and use the subscript to distinguish DNNs. Further, we abuse the subscript for a distribution and a DNN to reparameterise the corresponding random variable. For example, $p_\theta(x)$ is a distribution of x and $h_\theta(\cdot)$ is a DNN to reparameterise x, both with the parameter θ.

4.3 Probabilistic Iterative Networks

Now we develop the proposed computational framework, Probabilistic Iterative Networks (PIN). The goal is to learn a map from the observation y to an approximate posterior distribution $q(x|y)$. The map is modelled with a deep iterative network that recurrently inputs the forward map and observation and refines the reconstruction samples, and the probabilistic encoder therein enables generating diverse reconstruction samples. It employs the CVAE loss [126, 135], a conditional variant of variational autoencoders (VAEs) [80] (see [81] for a detailed introduction), and is trained using the reparameterisation trick. Below we first describe VAEs, the reparameterisation trick and the approximate *aleatoric* uncertainty interpretation with CVAEs [126, 141], and then develop the proposed framework.

4.3.1 VAE, Reparameterisation Trick and CVAE

Let $p_{\tilde{\theta}}(x|y)$ be the intractable target distribution of interest, where the vector $\tilde{\theta}$ in $p_{\tilde{\theta}}(x|y)$ contains the parameters of both prior $p_{\tilde{\theta}}(x)$ and likelihood $p_{\tilde{\theta}}(y|x)$, e.g., prior belief strength and noise precisions etc. In practice, the vector $\tilde{\theta}$ may be viewed as hyperparameters and estimated from the given data y simultaneously with the variational parameter $\tilde{\phi}$ in the approximation $q_{\tilde{\phi}}(x|y)$, e.g., by EM type algorithms.

To explore the target $p_{\tilde{\theta}}(x|y)$, variational Bayes is employed, which is a popular posterior approximation technique in machine learning [134]. It selects the best approximation $q_{\tilde{\phi}}(x|y)$ from a candidate family $\mathscr{Q}$ (parameterised by the vector $\tilde{\phi}$) by minimising suitable divergence, notably Kullback-Leibler (KL) divergence [83]. As reviewed in Chapter 1, the KL minimisation is often transformed into an equivalent evidence lower bound (ELBO) maximisation

$$\max_{\tilde{\phi}} \left\{ \mathscr{L}(\tilde{\theta}, \tilde{\phi}, y) = \mathbb{E}_{q_{\tilde{\phi}}(x|y)}[\log p_{\tilde{\theta}}(y|x)] - D_{\mathrm{KL}}(q_{\tilde{\phi}}(x|y)||p_{\tilde{\theta}}(x)) \right\}. \tag{4.1}$$

Solving the optimisation problem (4.1) requires evaluating the gradient of the functional $\mathscr{L}$ with respect to $\tilde{\phi}$, i.e., $\nabla_{\tilde{\phi}} \mathbb{E}_{q_{\tilde{\phi}}(x)}[f_{\tilde{\theta}}(x)]$ for a deterministic function $f_{\tilde{\theta}}$ parameterised by $\tilde{\theta}$. The challenge lies in the fact that the integral $\mathbb{E}_{q_{\tilde{\phi}}(x)}[f_{\tilde{\theta}}(x)]$ is often not analytically tractable and can only be evaluated by Monte Carlo methods. The reparameterisation trick [80, 116] (see the review [101] for other methods and their relative merits) is useful to overcome the challenge. It assumes that the conditional sampling of x depends on the condition y and an easy-to-sample auxiliary variable z distributed according to $p(z)$:

$$x = g_{\tilde{\phi}}(y, z), \quad \text{with} \quad z \sim p(z),$$

where $g_{\tilde{\phi}}$ is a deterministic function and can be modelled by DNNs. Then one obtains the following Monte Carlo estimator of (4.1): if the KL term is not available analytically,

$$\mathscr{L}^A = \frac{1}{L} \sum_{\ell=1}^{L} [\log p_{\tilde{\theta}}(y_i|x_{(i,\ell)}) - \log q_{\tilde{\phi}}(x_{(i,\ell)}|y_i)]$$

and if the KL term is available analytically,

$$\mathscr{L}^B = \frac{1}{L} \sum_{\ell=1}^{L} [\log p_{\tilde{\theta}}(y_i|x_{(i,\ell)})] - D_{\mathrm{KL}}(q_{\tilde{\phi}}(x|y_i)||p_{\tilde{\theta}}(x)),$$

where $\{x_{(i,\ell)}\}_{\ell=1}^{L}$ are L samples generated with y_i and using the variational encoder $q_{\tilde{\phi}}(x|y)$: $\{z_\ell\}_{\ell=1}^{L}$ are sampled from $p(z)$, and $x_{(i,\ell)} = g_{\tilde{\phi}}(y_i, z_\ell)$.

There are two neural networks in VAEs, an encoding network with parameter $\tilde{\phi}$ and a decoding network with parameter $\tilde{\theta}$. In practice, VAEs often do not use the encoding network to model the

parameterisation function $g_{\tilde{\phi}}$ in an end-to-end way. Instead, it takes the observation y and outputs the coefficients to reparameterise the unobservable x. In particular, for a multivariate Gaussian $q_{\tilde{\phi}}(x|y) = \mathcal{N}(x|\mu(y), \Sigma(y))$, the decoding network can output the mean $\mu(y)$ and Cholesky factor $L(y)$ of the covariance $\Sigma(y) = L(y)L(y)^T$. By sampling z from the standard Gaussian distribution $p(z) = \mathcal{N}(z|0, I)$, we can recover samples from $q_{\tilde{\phi}}(x|y) = \mathcal{N}(x|\mu(y), \Sigma(y))$ by

$$x = \mu(y) + L(y)z.$$

In theory, the decoding network can also output the full parameters to reparameterise the unobservable x. However, in practice, one usually employs an identity variance Gaussian with the mean being decoding output and leaves the prior distribution as the standard Gaussian. VAE allows performing both variational inference and model selection simultaneously, i.e., with respect to $\tilde{\phi}$ and $\tilde{\theta}$, respectively. Thus, the actual objective of VAEs is

$$\max_{\tilde{\phi},\tilde{\theta}} \{\mathscr{L}_{\text{VAE}}(\tilde{\theta}, \tilde{\phi}; y) = \mathbb{E}_{q_{\tilde{\phi}}(x|y)}[\log p_{\tilde{\theta}}(y|x)] - D_{\text{KL}}(q_{\tilde{\phi}}(x|y)||p_{\tilde{\theta}}(x))\}.$$

However, a direct application of VAEs to inverse problems is problematic: VAEs are unsupervised and use only noisy observations y (i.e., noisy observations of $\mathscr{A}(x)$) but not the ground-truth data x in the training. In the context of inverse problems, the use of the ground truth data x help tackle the intrinsic ill-posedness. To circumvent the issue, we employ the conditional VAEs (CVAEs) loss [126, 135]:

$$\max_{\phi,\theta} \left\{\mathscr{L}_{\text{CVAE}}(\theta, \phi; x, y) = \mathbb{E}_{q_\theta(z|x,y)}[\log p_\phi(x|y,z)] - D_{\text{KL}}(q_\theta(z|x,y)||p_\phi(z|y))\right\}. \tag{4.2}$$

There are three distributions: a teacher encoder $q_\theta(z|x,y)$, a student encoder $p_{\phi_1}(z|y)$ and a conditional decoder $p_{\phi_2}(x|y,z)$. The vector $\phi = (\phi_1, \phi_2)$ assembles the parameters of $p_{\phi_1}(z|y)$ and $p_{\phi_2}(x|y,z)$. Interestingly, the CVAE loss admits the following approximate inference interpretation [141].

Proposition 4.3.1. *Optimising $\mathscr{L}_{\text{CVAE}}(\theta, \phi; x, y)$ (expected on the training data distribution) is equivalent to optimising an upper bound of the expected reversed KL divergence*

$$J^*(p(x|y)) = \mathbb{E}_{p^*(y)}[D_{\text{KL}}(p^*(x|y)||p(x|y))].$$

Proof. See Appendix C.1 □

Thus, CVAEs indeed learn an optimal map from the observation y to an approximate posterior $p(x|y)$, in the sense of minimising expected loss of reversed KL divergence. This interpretation underpins the validity of the procedure for quantifying *aleatoric* uncertainties.

Below, we model the conditional encoder $p_{\phi_2}(x|y,z)$ by a mean-field Gaussian with variance βI (β is a hyperparameter). Then the DNN with parameter ϕ_2 only outputs the mean of $p_{\phi_2}(x|y,z)$, and on a mini-batch $\{(x_i,y_i)\}_{i=1}^M$, the objective function is given by:

$$\mathscr{L}^L_{\mathrm{CVAE}}(\phi,\theta) = -\frac{1}{2M}\sum_{i=1}^{M}\frac{1}{L}\sum_{\ell=1}^{L}\|x_i-\hat{x}_{(i,\ell)}\|^2 - \frac{\beta}{M}\sum_{i=1}^{M} D_{\mathrm{KL}}(q_\theta(z|x_i,y_i)||p_{\phi_1}(z|y_i)),$$

where $\hat{x}_{(i,\ell)}$ is the mean of $p_{\phi_2}(x|y_i,z_{i,\ell})$ and $\{z_{i,\ell}\}_{\ell=1}^L$ are L samples drawn from $q_\theta(z|x_i,y_i)$. In practice, we can reduce the computational complexity by letting $L=1$. Note that, for special choices of $q_\theta(z|x,y)$ and $p_{\phi_1}(z|y)$, the KL divergence term may be evaluated analytically, and can be used, if available. The gradient of the functional $\mathscr{L}^L_{\mathrm{CVAE}}(\phi,\theta)$ is then evaluated by the reparameterisation trick.

Remark 4.3.1. *In VAEs, the approximate posterior of the unobservable x is modelled by $q_\phi(x|y)$, whereas in CVAEs, it is modelled by $p_\phi(x|y)=\int p_{\phi_2}(x|y,z)p_{\phi_1}(z|y)dz$. In both VAEs and CVAEs, the stochasticity of x comes from z: in VAEs, z is independent on the observation y, whereas in CVAEs, z is dependent on y. Since the distribution of z is learned, it is more flexible than that in VAEs. Thus, even if $p_{\phi_2}(x|y,z)$ is chosen to be simple distributions, e.g., Gaussian distributions, $p_{\phi_1}(x|y)$ can still model a broad family of distributions for continuous unobservable x due to the presence of $p_{\phi_1}(z|y)$, in a manner similar to scale mixture of Gaussians [141].*

4.3.2 Probabilistic Iterative Networks (PIN)

A probabilistic modelling framework consists of one learning principle given by a loss function with proper probabilistic interpretation, and one graphical model describing probabilistic dependency between variables. In the proposed framework, we employ a CVAE type loss function:

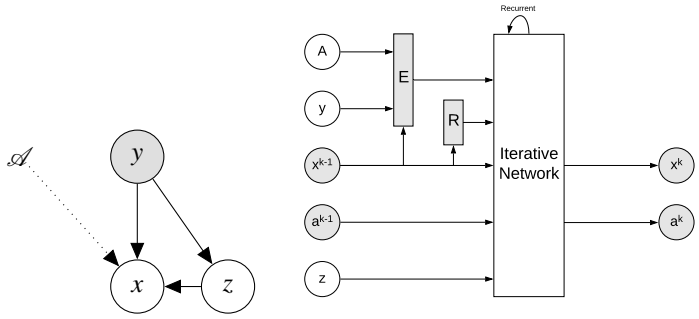
$$\max_{\phi,\theta}\Big\{\mathscr{L}_{\mathrm{PIN}}(\theta,\phi;x,y,\mathscr{A}) = \mathbb{E}_{q_\theta(z|x,y,\mathscr{A})}[\log p_\phi(x|y,z,\mathscr{A})] - D_{\mathrm{KL}}(q_\theta(z|x,y,\mathscr{A})||p_\phi(z|y))\Big\}. \quad (4.3)$$

Its difference from the standard CVAE loss (4.2) is that (4.3) also includes the forward map $\mathscr{A}$ and its adjoint $\mathscr{A}^*$ as training data. Here $\mathscr{A}$ and $\mathscr{A}^*$ may have different realisations (e.g., corresponding to different discretisations) with varying dimensions. Nonetheless, it is a deterministic variable: during the training for one class of inverse problems, the maps are all discretisations of the underlying continuous operator. Then the modelled approximate posterior $p_\phi(x|y)$ is given by

$$p_\phi(x|y) = \int p_\phi(x|y,z,\mathscr{A})p_\phi(z|y)\mathrm{d}z.$$

It is summarised by the graphical model in Fig. 4.1(a), where shaded and non-shaded nodes represent observations and hidden variables, respectively, and solid and dotted arrows denote probabilistic dependencies and explicit incorporations, respectively. Lying at the core of the framework is a learning algorithm that can learn a conditional sampler, in a manner similar to a random number

generator (RNG) for a given distribution. Note that for an RNG, different runs lead to different samples, but with a fixed random seed, it repeats the path for different runs. The auxiliary (low-dimensional) latent variable z conditioned on the observation y is an analogue of the random seed in the RNG, and is introduced into the deterministic iterative process (modelled by an iterative network) to diversify the reconstruction samples. In particular, for a fixed z, the iteration process inputs the sample initialisation and applies a recurrent refining step based on suitable sample quality measure and the information encoded in the auxiliary variable z.



(a) graphical model (b) iterative network

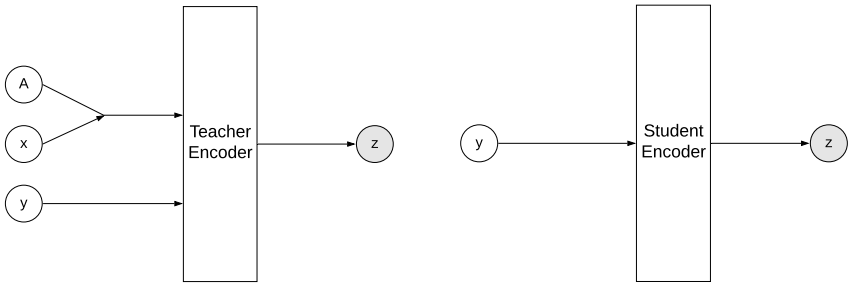
Figure 4.1: The graphical model and iterative network of the proposed framework.

Three DNNs are employed to model the distributions in the loss function $\mathscr{L}_{\text{PIN}}(\theta,\phi;x,y,\mathscr{A})$ and used for the conditional sampling process, including one for the iterative process, i.e., iterative network, and two for probabilistic encoders, i.e., teacher encoder and student encoder. The observation y and forward map $\mathscr{A}$ constitute their inputs: the observation y is input into the two probabilistic encoders and also each recurrent of the iterative network; the forward map is also input into each recurrent of the iterative network and also in one of the probabilistic encoders (i.e., the teacher encoder) during training. Below we explain how the three networks work separately.

The recurrent component is the (deep) iterative network $h_{\phi_2}(\cdot)$. See Fig. 4.1(b) for a schematic illustration, where shaded and nonshaded circles denote variables for updating and fixed variables during the sampling process, respectively, and shaded and nonshaded rectangles represent functional input to the network and the iterative network, respectively. During the process of one sampling, only the values of shaded circles and rectangles change. The network begins with the initial guess x^0 (default: backprojected data $\mathscr{A}^*y$) and outputs x^K after K iterations as the mean of $p_\phi(x|y,z,\mathscr{A})$. At the k-th iteration, the network takes one sample x^{k-1} to refine and outputs an improved sample x^k. To incorporate the forward map $\mathscr{A}$ and the observation y, we employ a functional $E(\mathscr{A},y,x^{k-1})$. In the lens of variational regularisation [67], $E(\mathscr{A},y,x^{k-1})$ measures how well x^{k-1} can explain the data y. To indicate how well x^{k-1} fulfils the prior knowledge, e.g., sparsity (in a transformed domain), we use also the penalty $R(x^{k-1})$ as a part of the input. For the sample quality measure $E(\mathscr{A},y,x^{k-1})$ and the penalty $R(x^{k-1})$, we use $||y-\mathscr{A}(x^{k-1})||^2$ (or its gradient), and $||x^{k-1}||_2^2$ or

$|x^{k-1}|_{\mathrm{TV}}$ (total variation semi-norm), respectively. Besides the latest iterate x^{k-1} and the quality indicators E and R, the network $h_{\phi_2}(\cdot)$ also takes a memory variable a^{k-1} and an auxiliary variable z. The memory variable a^{k-1} plays the role of momentum in gradient type methods and is to retain long-term information between iterations. The auxiliary random variable z is low-dimensional and to introduce randomness into the iteration procedure. Since both x^{k-1} and x^k belong to the unobservable space $\mathbb{R}^n$, we adopt CNNs without pooling layers to model the iterative network. Different inputs of $h_{\phi_2}(\cdot)$ are concatenated along the channel axis, and the outputs of $h_{\phi_2}(\cdot)$, i.e. the update δx^k (with $x^k = x^{k-1} + \delta x^k$) and the updated memory a^k, are also concatenated.

Remark 4.3.2. *Note that the proposed framework re-uses the observation data y and the maps $\mathscr{A}$ and $\mathscr{A}^*$ for refinement at each step, and the overall procedure differs greatly from the conventional deterministic mapping that serves as a post-processing step of back-projection. The latter are end-to-end mappings that take the backprojected data and output a refinement, whereas PIN employs the current sample and the quality measures, and decide the refinement strategy accordingly.*



(a) teacher encoder $q_\theta(z|x,y,\mathscr{A})$ (b) student encoder $p_{\phi_1}(z|y)$

Figure 4.2: Probabilistic encoders in the framework. Shaded nodes denote the random variable.

The framework also involves two encoders of z, i.e., a teacher encoder $q_\theta(z|x,y,\mathscr{A})$ and a student encoder $p_{\phi_1}(z|y)$, and with the reparameterisation trick, both encoders output coefficients to reparameterise the random variable z; see Fig. 4.2 for a schematic illustration. The student encoder $p_{\phi_1}(z|y)$ takes the observation y, and encodes the observation-based knowledge so as to inform the iterative network. Given one sample z from $p_{\phi_1}(z|y)$, the iterative network gives one refining increment, and the distribution of z contributes to the diversity of the unobservable x. To help train the student encoder $p_{\phi_1}(z|y)$, we input the ground truth x and the forward map $\mathscr{A}$ into the teacher encoder $q_\theta(z|x,y,\mathscr{A})$. The teacher encoder $q_\theta(z|x,y,\mathscr{A})$ is discarded once the training is finished and only the student encoder $p_{\phi_2}(z|y)$ is used at the inference stage. The encoders $p_{\phi_1}(z|y)$ and $q_\theta(z|x,y,\mathscr{A})$ are modelled by two DNNs $h_{\phi_1}(\cdot)$ and $h_\theta(\cdot)$, respectively, which reparameterise the auxiliary variable z. Since the variable z is low-dimensional with a predetermined dimension, CNNs with reduced mean layers and 1×1 convolutional layers can guarantee the dimension flexibility of the input y. To input the ground-truth data of x with y into $h_\theta(\cdot)$ with the dimension flexibility, we

use the forward map $\mathscr{A}$ and concatenate $\mathscr{A}(x)$ with y along the channel axis. It is worth noting that the framework is very flexible with the problem dimension, and one can conduct training (x_i, y_i) of different shapes (and the corresponding $\mathscr{A}_i$) .

Now we can state the algorithms for training and inference of PIN, cf. Algorithms 5 and 6, respectively. In the algorithms, M denotes the mini-batch size, T the maximum number of training batches, K the number of recurrences of h_{ϕ_2} for one sample, and $(\hat{\phi}, \hat{\theta})$ the output of the training (i.e., the learned parameters). There are many possible choices of the stochastic optimiser at line 11 of Algorithm 5, e.g., ADAM, and SGD. We shall employ ADAM in our experiment. The final sample from the iterative process is regarded as the mean of the conditional distribution $p_\phi(x|y,z,\mathscr{A})$. Thus, given the initial x^0, the iteration with different realisations of z leads to diverse samples of the unobservable x. Since each sample is the mean of some conditional distribution $p_\phi(x|y,z,\mathscr{A})$ rather than a direct sample from the target approximate posterior, the summarising statistics should be transformed into that of the target distribution as follows. The posterior variance contains two components: one is due to the background (i.e., βI), and the other is due to the sample variance (i.e., $\frac{1}{S}\sum_{i=1}^{S} x_i x_i^t - \widehat{\mathbf{E}}_{p(x|y)}[x]\widehat{\mathbf{E}}_{p(x|y)}[x]^t$).

Proposition 4.3.2. *Let the approximate posterior $p_\phi(x|y) = \int p_\phi(x|y,z,\mathscr{A})p_\phi(z|y)dz$ be a mixture of Gaussian distributions, i.e., $p_\phi(x|y,z,\mathscr{A}) = \mathscr{N}(x|x^K(z),\beta I)$, and z be the mixture variable. Then given samples $\{z_i\}_{i=1}^S$ of z from $p_\phi(z|y)$, and the corresponding $x^K(z)$, denoted by $\{x_i\}_{i=1}^S$, the mean $\mathbf{E}_{p(x|y)}[x]$ and the covariance $\mathbf{Cov}_{p(x|y)}[x]$ of $p_\phi(x|y)$ can be estimated by the following unbiased estimators:*

$$\widehat{\mathbf{E}}_{p(x|y)}[x] = \frac{1}{S}\sum_{i=1}^{S} x_i \quad \textit{and} \quad \widehat{\mathbf{Cov}}_{p(x|y)}[x] = \beta I + \frac{1}{S}\sum_{i=1}^{S} x_i x_i^t - \widehat{\mathbf{E}}_{p(x|y)}[x]\widehat{\mathbf{E}}_{p(x|y)}[x]^t. \tag{4.4}$$

Proof. For the mean $\mathbf{E}_{p(x|y)}[x]$, by dentition, there holds

$$\begin{aligned}\mathbf{E}_{p(x|y)}[x] &= \int_x x p(x|y)\mathrm{d}x = \int_x x \int_z p_\phi(x|y,z,\mathscr{A})p_\phi(z|y)\mathrm{d}z\mathrm{d}x \\ &= \int_z \int_x x p_\phi(x|y,z,\mathscr{A})\mathrm{d}x p_\phi(z|y)\mathrm{d}z = \int_z x^K(z)p_\phi(z|y)\mathrm{d}z.\end{aligned}$$

Thus, $\frac{1}{S}\sum_{i=1}^{S} x_i$ is an unbiased estimator of $\mathbf{E}_{p(x|y)}[x]$. Similarly, for the covariance $\mathbf{Cov}_{p(x|y)}[x]$, by the standard bias variance decomposition,

$$\mathbf{Cov}_{p(x|y)}[x] = \int xx^T p(x|y)\mathrm{d}x - \mathbf{E}_{p(x|y)}[x]\mathbf{E}_{p(x|y)}[x]^T$$

Now the first term on the right hand side, there holds

$$\begin{aligned}\int xx^T p(x|y)\mathrm{d}x &= \int_x xx^T \int_z p_\phi(x|y,z,\mathscr{A}) p_\phi(z|y)\mathrm{d}z\mathrm{d}x = \int_z \int_x xx^T p_\phi(x|y,z,\mathscr{A})\mathrm{d}x p_\phi(z|y)\mathrm{d}z \\ &= \int_z (\mathbf{Cov}_{p_\phi(x|y,z,\mathscr{A})}[x] + \mathbf{E}_{p_\phi(x|y,z,\mathscr{A})}[x]\mathbf{E}_{p_\phi(x|y,z,\mathscr{A})}[x]^T) p_\phi(z|y)\mathrm{d}z \\ &= \beta I + \int_z \mathbf{E}_{p_\phi(x|y,z,\mathscr{A})}[x]\mathbf{E}_{p_\phi(x|y,z,\mathscr{A})}[x]^T p_\phi(z|y)\mathrm{d}z\end{aligned}$$

Consequently, $\beta I + \frac{1}{S}\sum_{i=1}^S x_i x_i^t - \widehat{\mathbf{E}}_{p(x|y)}[x]\widehat{\mathbf{E}}_{p(x|y)}[x]^t$ is an unbiased estimator of $\mathbf{Cov}_{p(x|y)}[x]$. □

Algorithm 5 PIN training

Input: Training data $\{(\mathscr{A}_i, x_i, y_i)\}_{i=1}^N$, β, T, K, M
for $t = 1, 2, \ldots, T$ **do**
  Randomly select a mini-batch training data $\{(\mathscr{A}_i, x_i, y_i)\}_{i=1}^M$;
  Sample $\{z_i\}_{i=1}^M$ from $\{q_\theta(z|x_i, y_i)\}_{i=1}^M$;
  Initialise $\{\hat{x}_i\}_{i=1}^M$ with $\{\mathscr{A}^*(y_i)\}_{i=1}^M$ and $\{a_i\}_{i=1}^M$ with zeros;
  for $k = 1, 2, \ldots, K$ **do**
    Update $\{\hat{x}_i\}_{i=1}^M$ and $\{a_i\}_{i=1}^M$ with $\{h_{\phi_2}(\hat{x}_i, E(\mathscr{A}_i, y_i, \hat{x}_i), R(\hat{x}_i), a_i, z_i)\}_{i=1}^M$;
  end for
  Evaluate the KL divergence $\{D_{\mathrm{KL}}(q_\theta(z|x_i, y_i)||p_{\phi_1}(z|y_i))\}_{i=1}^M$;
  Compute the objective function $\mathscr{L}^1_{\mathrm{CVAE}}(\phi, \theta)$;
  Update the parameters (ϕ, θ);
end for
Output: $(\hat{\phi}, \hat{\theta})$

Algorithm 6 PIN inference

Input: Test data $(\mathscr{A}, y)$, S, K, $\hat{\phi} = (\hat{\phi}_1, \hat{\phi}_2)$
for $s = 1, 2, \ldots, S$ **do**
  Sample z_s from $p_{\phi_1}(z|y)$
  Initialise $\hat{x}_s$ with $\mathscr{A}^*(y)$ and a with zeros
  for $k = 1, 2, \ldots, K$ **do**
    Update $\hat{x}_s$ and a with $h_{\phi_2}(\hat{x}_s, E(\mathscr{A}, y, \hat{x}_s), R(\hat{x}_s), a, z_s)$
  end for
end for
Output: $\{\hat{x}_s\}_{s=1}^S$
Evaluate $(\widehat{\mathbf{E}}_{p(x|y)}[x], \widehat{\mathbf{Cov}}_{p(x|y)}[x])$ by Eq. (4.4)

Remark 4.3.3. *Generative modelling frameworks do not naturally guarantee dimension flexibility, due to the constraints induced by DNN architectures connecting the variables, and are not flexible when using only the training data* $\{(x_i, y_i)\}_{i=1}^N$. *Also recent approaches [4] use the unobservable and backprojected data pairs* $\{(x_i, \mathscr{A}_i^*(y_i))\}_{i=1}^N$, *which is potentially at the expense of a loss of information.*

Remark 4.3.4. *In the machine learning literature, the idea of learning an algorithm is often referred to as Learning to Learn (L2L) [10, 29, 46]. It aims at extracting meta knowledge for a class of similar problems. PIN represents a probabilistic extension of L2L: by introducing an auxiliary variable z,*

it enables the iterative network to output samples from a modelled distribution, thereby extending L2L to generative modelling. Meanwhile, in L2L, the sample quality measure is the same as the loss function, whereas in PIN, the objective function is the CAVE loss, which is different from E or R. That is, PIN does not minimise any deterministic functional $E(\cdot)+\lambda R(\cdot)$, and instead it learns how to makes use of the information during the training process. In particular, it does not assume any noise type of observation y by using the measure $E(\cdot)$.

4.4 Numerical Experiments and Discussions

In this part we showcase the proposed computational framework with experiments on positron emission tomography (PET). It is a pillar of modern diagnostic imaging, allowing noninvasive, sensitive and specific detection of functional changes in a number of diseases. As reviewed in Chapter 1, most PET reconstruction algorithms rely on penalised maximum likelihood estimates, using a hand crafted prior (e.g., total variation and anatomical priors) [111], or more recently learning based approaches, e.g., unrolled deep iterative networks, have been proposed. While these techniques have been successful, they lack the capability to provide uncertainty estimates. Thus, it is of much interest to provide uncertainties to relevant point estimates.

For the experiments, we employ a 3-layer CNN as the iterative network h_{ϕ_2} and fix $K = 10$ iterations for each sampling step, cf. Fig. 4.3, and VGG style encoders for both h_{ϕ_1} and h_θ, cf. Fig. 4.4. We train PIN on a synthetic dataset consisting of elliptical phantoms [3], and test it on the dataset BrainWeb [30]. Throughout, the training pair $(x,y) \in \mathbb{R}^{128\times128} \times \mathbb{R}^{30\times183}$, and the forward map is the Radon transform, which is normalised, and different peak values of x are used to indicate the count level: 1e4 and 1e2 for respectively moderate and low count levels. The observation y is generated by corrupting the sinogram $\mathscr{A}x$ by Poisson noise, i.e., $y_i \sim \text{Pois}((\mathscr{A}x)_i)$. The hyper-parameter β is tuned in a trial-and-error manner, and fixed at 5e-3 below. The experiments are conducted on a desktop with two Nvidia GeForce 1080 Ti GPUs and Intel i7-7700K CPU 4.20GHz×8. PIN is trained for $T =$ 1e5 batches, each of which contains 10 randomly generated (x,y) pairs on the fly. The training almost converges after 2e4 batches and it takes around 11 hours to go over all 1e5 batches. The summarising statistics reported below are computed from 1000 samples for each observation y generated by the trained PIN. It takes averagely 12.66s to sample 1000 images of size 128×128 from a trained framework. The implementation uses the following public deep learning frameworks: `Tensorflow` [1], `Tensorflow Probability` [36], `DeepMind Sonnet` (`https://github.com/deepmind/sonnet`) and `ODL` (`https://github.com/odlgroup/odl`), and the source code can be found at `https://github.com/chenzxyz/prob_iterative_net`.

Remark 4.4.1. *To validate that our framework can be trained on one dimensional setting and used on different dimensional settings, we only incorporate one realisation (matrix form) of the forward operator $\mathscr{A}$ per training. For example, we fixed the number of angles as* 30 *and fixed the number*

of projection per angle as 183 *for training data with x of size* 128×128. *Note that for different realisations of the forward operator, the null space of the matrices can be different. The discussion of the generalisation of one trained framework on test data with operator realisation deviation is interesting but not of the main focus on this study. As a result, we leave it as possible future work. Nevertheless, this problem could be mitigated by incorporating various operator realisations covering different null spaces during single training, due to the flexibility of our framework.*

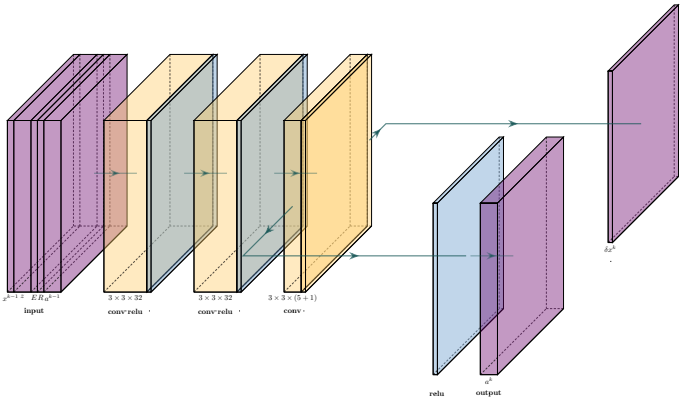


Figure 4.3: The layer configuration of the iterative network h_{ϕ_2}: $3 \times 3 \times 32$ denotes convolutional layer with a kernel size 3×3 and 32 output channels. In the third convolutional layer, $5+1$ denotes 5 channels for memory a^k and 1 channel for the update δ^k.

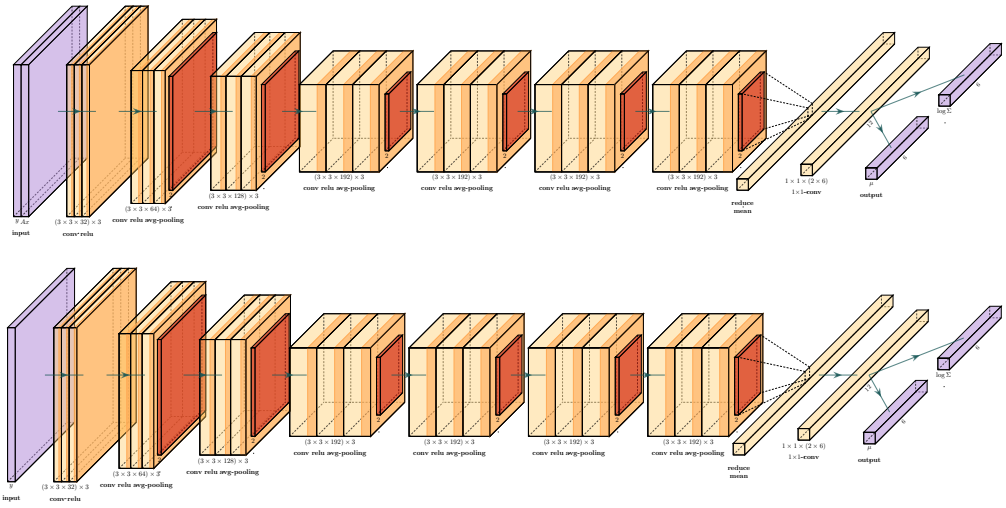


Figure 4.4: The layer configurations of the teacher encoder (top) and student encoder (bottom): $(3 \times 3 \times 32) \times 3$ denotes 3 convolutional layers respectively followed by an ReLU layer with a kernel size 3×3 and 32 output channels. 2 under the brown layer denote average pooling layer with stride size 2. $1 \times 1 \times (2 \times 6)$ denotes 1×1 convolutional layer with 12 output channels, i.e. 6 for mean μ and 6 for log (diagonal) variance $\log \Sigma$.

4.4.1 Flexibility of PIN

The proposed PIN is very flexible and is applicable to a broad range of inverse problems. In this part, we illustrate the following distinct features (i) transferability on different datasets and (ii) flexibility with respect to problem dimension. To see (i), we show several samples from the training dataset, which consists of elliptical phantoms, and more realistic medical images (BrainWeb) in Fig. 4.5, for two peak values 1e2 and 1e4. In the low-count case, the backprojection $\mathscr{A}^* y$ is very noisy, and represents a poor approximation to the ground truth $x^\dagger$ and thus it is numerically far more challenging for image reconstruction than the moderate count case. Visually, the phantoms in the training data do not resemble real medical images, due to their lack of fine detailed structures. Nonetheless, the mean of the approximate posterior represents excellent reconstructions on the BrainWeb dataset; see rows 2–4 of Fig. 4.6 for exemplary results on ten test images from BrainWeb, where the training and test data have identical dimensions. The posterior variance, computed according to (4.4), captures the overall shape of phantom, similar to that obtained by expectation propagation [140], and the overall shape also resembles the difference between the posterior mean and ground truth but with less detailed structures. This is characteristic of PET reconstructions, where the reconstructed images often lack very fine details. In summary, PIN can learn high level knowledge from the training data (less close to realistic medical images) to facilitate the reconstructions, and performs well on test data (closer to realistic medical images), and the explicit incorporation of the forward operators allows disentangle the mixture of unobservable and observation spaces so as to avoid overfitting synthetic training data.

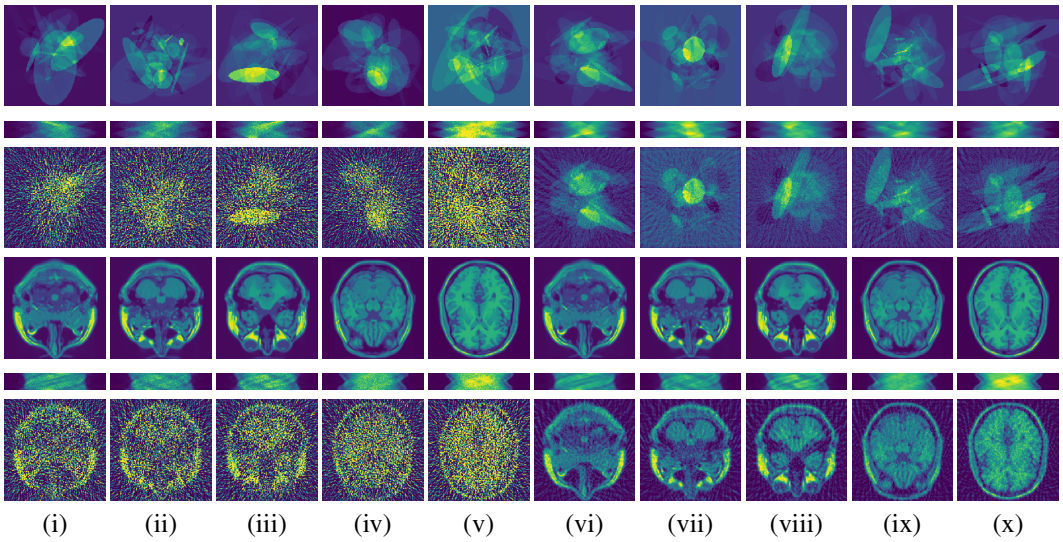


Figure 4.5: Samples of synthetic training data (row 1–3) and test data from BrainWeb (row 4–6): ground truth phantoms $x^\dagger$, noisy sinograms y and backprojected data $\mathscr{A}^*(y)$: (i)–(v) and (vi)–(x) refer to low and moderate count levels, respectively.

Now we illustrate the feature (ii), i.e., dimension flexibility of PIN, with one trained PIN with the training data $(x,y) \in \mathbb{R}^{128\times128} \times \mathbb{R}^{30\times183}$, peak value of x at 1e4, and test the flexibility by

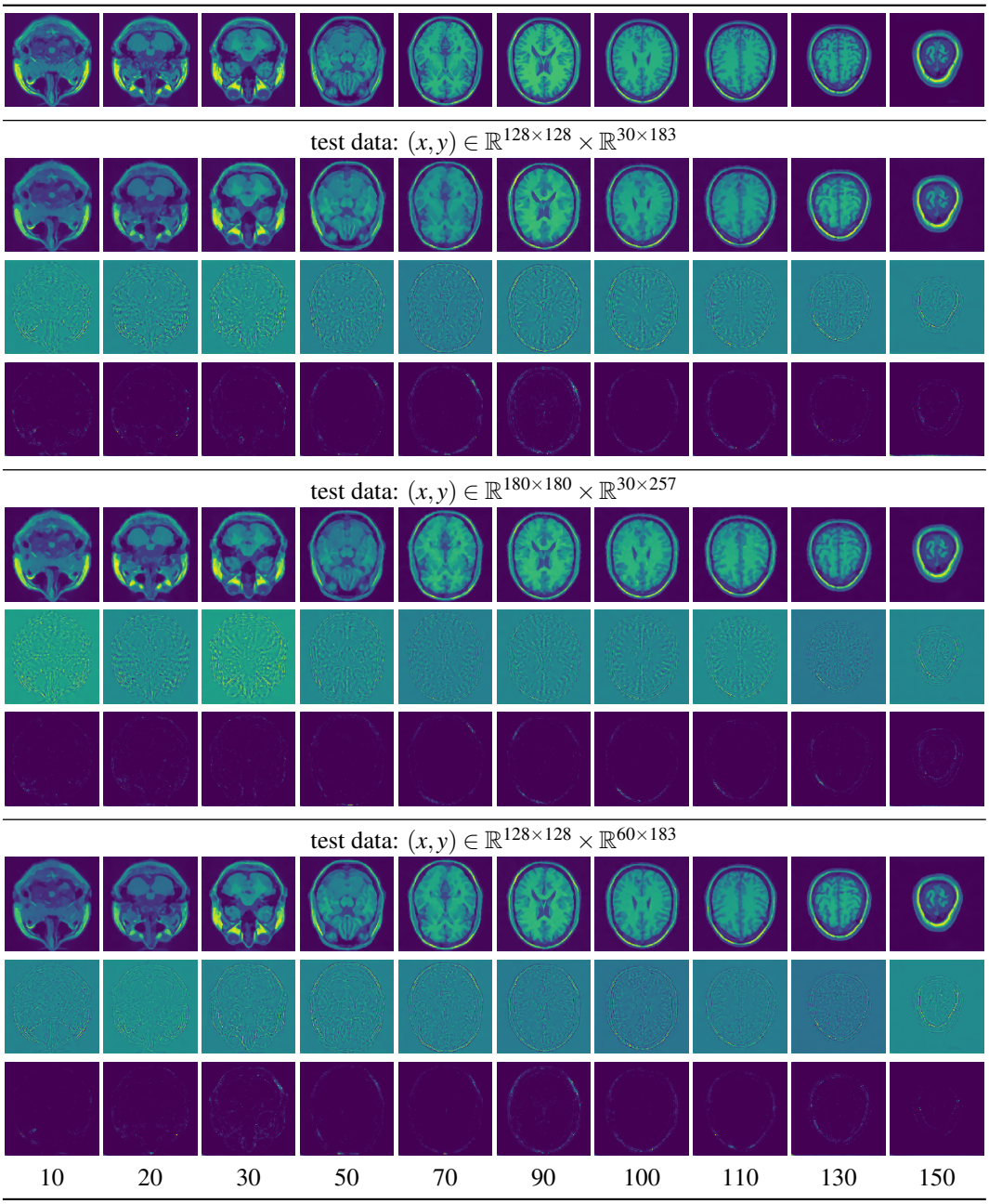


Figure 4.6: Reconstructions of 10 samples from BrainWeb with peak value 1e4 by PIN. The top row denotes ground truth phantoms. Rows 2–4, 5–7, and 8–10 are for test data of size $(x,y) \in \mathbb{R}^{128\times128} \times \mathbb{R}^{30\times183}$, $(x,y) \in \mathbb{R}^{180\times180} \times \mathbb{R}^{30\times257}$ and $(x,y) \in \mathbb{R}^{128\times128} \times \mathbb{R}^{60\times183}$, respectively. Within each block, from top to bottom: posterior mean $\hat{x}$, the difference $\hat{x} - x^{\dagger}$, and posterior variance.

varying the dimension of either x or y. First, we fix the number of project angles in the forward map and increase the dimension of x by using $(x,y) \in \mathbb{R}^{180\times180} \times \mathbb{R}^{30\times257}$; see rows 5–7 of Fig. 4.6 for the corresponding reconstructions. It is observed that PIN can be trained on a low-dimensional dataset and still perform well on high-dimensional data. Second, we fix the dimension of x and increase the number of projections in the forward map using $y \in \mathbb{R}^{60\times183}$. The reconstructions in rows 8–10 of Fig. 4.6 show that although the training process has access to only observations

with more limited-angles, PIN can still give good reconstructions for more informative data. In either case, the observation on the variance structure remains largely valid. Table 4.1 summarises the corresponding quantitative results using two standard image quality measures, i.e., PSNR and SSIM [65], and the table also includes results for the same test images with a peak value 1e2. For either peak value, the image quality measures improve steadily as the number of projection angle increases or the dimension of x increases, but the latter has more pronounced effect. Further, the reconstruction accuracy degrades significantly as the peak value decrease from 1e4 to 1e2, where the latter is numerically far more challenging.

These features of PIN have important practical implications. First, PIN enables using synthetic datasets for training, and avoid expensive physically derived labeled training data, which is often expensive to collect. Second, PIN can be trained efficiently. Although the DNN feedforward propagation is efficient, the training is generally time consuming. One the key influencing factor is the data size. Our experiments indicate that PIN can be trained on a small dataset and then used on datasets of larger sizes, thereby reducing the training time. Third, PIN can extract high-level useful knowledge for image reconstruction, to some extent. Even though it has never seen the larger spectrum of projection angles during training, it still performs well on more informative data (i.e., with more angles). This is possibly due to explicit inclusion of the forward operators and thus the built model is more comprehensive.

Table 4.1: PSNR and SSIM values for the reconstructions by the trained PIN on ten phantoms with peak value 1e4 (MC) and 1e2 (LC), using different test data sizes. The column index refers to `Python` style index of the phantom in the BrainWeb dataset.

index		10	20	30	50	70	90	100	110	130	150
		PSNR									
(128,30)	MC	27.79	27.12	27.16	27.34	25.55	24.92	26.78	27.74	27.94	30.88
	LC	22.86	21.97	22.06	22.10	20.77	20.68	21.49	22.21	22.58	26.16
(180,30)	MC	29.49	28.92	28.69	29.11	28.14	27.28	28.29	29.29	29.59	33.30
	LC	23.59	23.57	23.40	23.54	23.01	22.67	23.19	23.52	24.08	27.62
(128,60)	MC	28.68	28.29	28.17	28.54	26.28	26.21	27.99	28.48	28.73	31.21
	LC	22.82	22.38	22.46	22.26	20.91	20.92	21.61	22.29	22.58	26.20
		SSIM									
(128,30)	MC	0.92	0.92	0.93	0.92	0.91	0.91	0.93	0.94	0.95	0.98
	LC	0.75	0.72	0.75	0.68	0.63	0.71	0.71	0.71	0.75	0.87
(180,30)	MC	0.93	0.93	0.93	0.93	0.93	0.92	0.93	0.94	0.96	0.98
	LC	0.77	0.75	0.76	0.73	0.72	0.73	0.74	0.73	0.77	0.88
(128,60)	MC	0.94	0.95	0.95	0.94	0.93	0.94	0.96	0.96	0.96	0.98
	LC	0.76	0.75	0.76	0.69	0.66	0.71	0.72	0.73	0.74	0.87

4.4.2 Comparison with Benchmarks

Now we compare PIN with conventional and deep learning based methods on all 181 phantoms in the BrainWeb dataset. PIN is compared with the following benchmark methods, i.e., maximum

likelihood EM (MLEM) [125], maximum a posteriori with total variation prior with nonnegativity constraint (TV-MAP) [64] and iterative deep neural network (IDNN) [3], and the results are summarised in Table 4.2. MLEM and TV-MAP are two of the most established iterative reconstruction methods in the PET community, and IDNN is an unrolled iterative method inspired by classical variational regularisation and exploits DNNs for iterative refinement. For MLEM, we use `odl` inbuilt solver `mlem`, and for TV-MAP, use the primal dual hybrid gradient method (implemented by `odl.solvers.pdhg`). The regularisation parameter for total variation prior is fixed at 2e-1 and 2e0 for the moderate and low count level, respectively, which was determined by a trial-and-error manner. The comparative results are summarised in Table 4.2, shown with SSIM and PSNR, averaged over all 181 phantoms in the BrainWeb dataset. The results clearly show that PIN can deliver state-of-the-art point estimates in terms of PSNR and SSIM, especially in the low count case. However, PIN additionally can provide uncertainty information which is unavailable from benchmark methods.

Table 4.2: Comparisons between PIN mean and benchmark methods on 181 BrainWeb phantoms at two count levels: 1e4 (MC) and 1e2 (LC).

	MLEM	TV-MAP	IDNN	PIN
MC	0.73/23.20	0.85/28.76	0.92/29.07	0.91/28.01
LC	0.64/21.55	0.62/22.58	0.59/21.68	0.63/23.10

To shed further insight, we evaluate the methods on phantoms with an artificially added tumour by changing the pixel values to the peak value. We (randomly) choose two phantoms from BrainWeb dataset (`Python` style index: 10 and 110). A small tumour of radius 2 and a large tumour of radius 5 are added into the 10-th phantom and the 110-th phantom, respectively. The corresponding reconstructions are shown in 4.7. It is observed the tumours can be clearly reconstructed by the PIN means for both count levels, except the small tumour at low count levels. In the latter case, none of the methods can reasonably reconstruct the tumour, due to too noise data compared to the signal strength. The results by PIN, IDNN and GM3 (to be described below) are comparable, at least visually. The ability of reconstructing tumours further indicates that PIN does not miss out important features non-present in the training data, as long as the signal is strong, since many machine learning based methods tend to miss the tumour due to the bias induced by tumour-free training data [100].

Next we compare PIN with the probabilistic approach [85], which reports state-of-the-art performance for *aleatoric* uncertainty. It employs (non-Bayesian) neural network ensembles to estimate predictive uncertainty, each network in the ensemble learn similar values close to the training data, and different ones in regions of the space far from the training data. Thus, the approach lacks rigorous Bayesian interpretation as PIN. To this end, we train a mixture with three multivariate Gaussians (GM3) without adversarial samples, where the training of each component of the mixture is to fit a mean network and a variance network using Gaussian log-likelihood to the data [102]. To stabilise the training procedure, we first train the mean network and then train the variance network. Alterna-

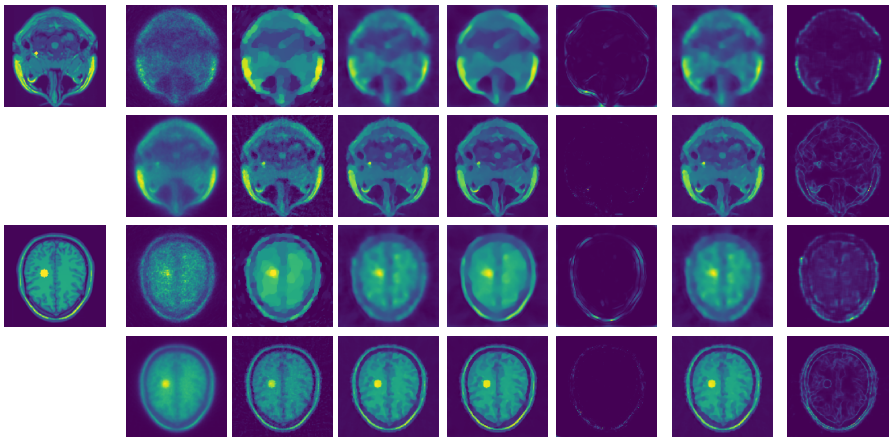


Figure 4.7: Tumour tests on two BrainWeb phantoms of compared benchmarks and PIN. For each phantom, the top row is for the low count level and the bottom row is for the moderate count level.

tively, one can train the mean network as a warmup and then train the mean and variance networks simultaneously, but this procedure usually leads to worse results, and thus we do not present the relevant results. The comparative quantitative results are given in Table 4.3. It is observed that PIN can provide better point estimates in terms of both SSIM and PSNR, which concurs with Fig. 4.8 in the low count case, whereas in the moderate count case, GM3 can sometimes deliver better results. This is consistent with the prevailing empirical observation that Bayesian type deep learning techniques tend to compromise the accuracy of prediction [104]. In terms of the variance map, that by GM3 contains more structural patterns and in particular resembles more closely the error map.

Table 4.3: PSNR and SSIM values for the reconstructions by the trained PIN and GM3 on ten phantoms with peak value 1e4 (MC) and 1e2 (LC). The column index refers to `Python` style index of the phantom in the BrainWeb dataset.

index		10	20	30	50	70	90	100	110	130	150
		PSNR									
PIN	MC	27.66	27.14	27.25	27.25	25.65	24.98	26.91	27.81	27.96	30.86
	LC	22.60	22.09	22.30	22.14	22.87	22.52	21.39	22.22	22.48	25.78
GM3	MC	28.05	27.48	27.43	27.50	26.77	26.83	27.74	29.33	32.57	28.21
	LC	21.86	21.35	21.09	20.69	19.32	19.02	19.74	20.61	21.32	23.67
		SSIM									
PIN	MC	0.92	0.92	0.93	0.92	0.91	0.91	0.93	0.94	0.95	0.98
	LC	0.74	0.74	0.76	0.67	0.62	0.68	0.71	0.72	0.74	0.86
GM3	MC	0.92	0.92	0.93	0.91	0.92	0.93	0.93	0.95	0.98	0.93
	LC	0.72	0.70	0.70	0.62	0.57	0.62	0.65	0.67	0.96	0.80

To shed more insights into the variance by PIN and the probabilistic benchmark GM3, we show the cross-section plots with marginal 0.95 posterior credible intervals in Fig. 4.9 for both moderate and low count levels. According to Proposition 4.3.2, the estimated variances by PIN contains two distinct sources, i.e., sample variance and the variance β of the conditional Gaussians

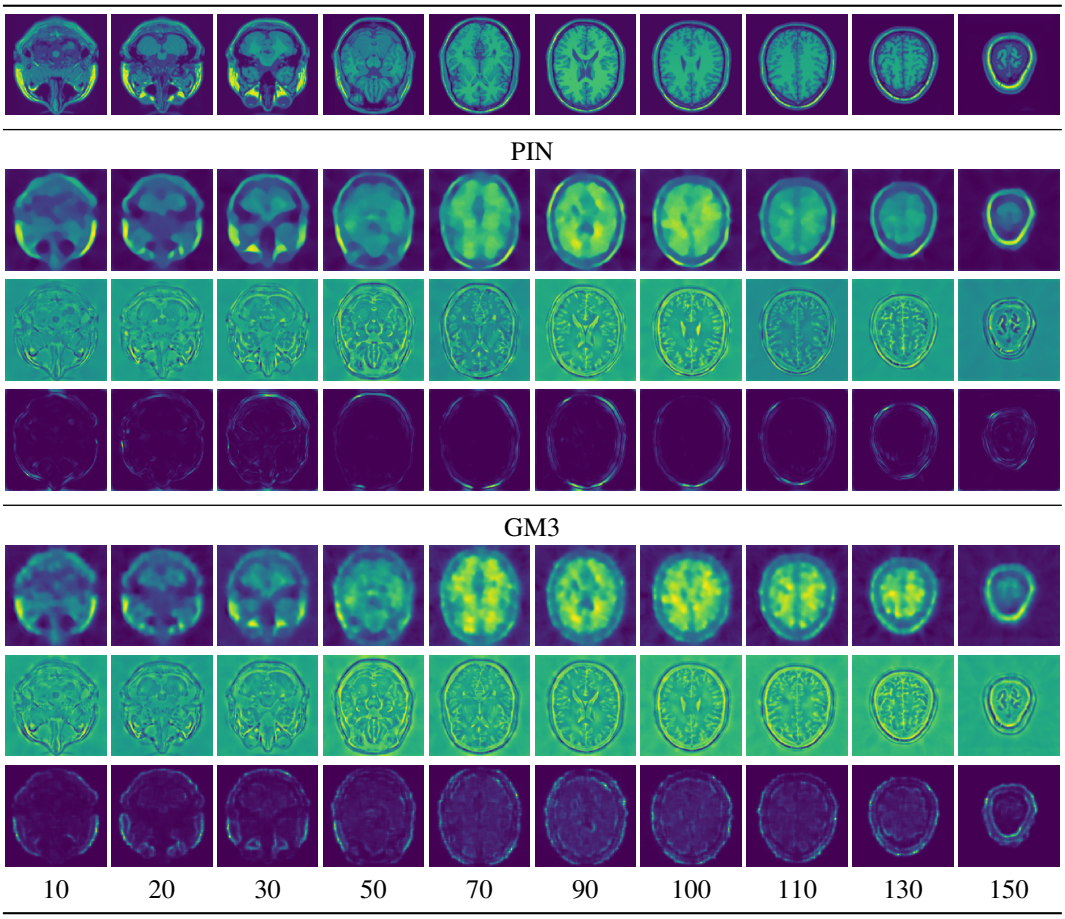


Figure 4.8: Reconstructions of 10 samples from BrainWeb with peak value 1e2. The top row refers to ground truth phantoms. The 2nd–4th and 5th–7th rows are results by PIN and GM3, respectively, from top to bottom: posterior mean $\hat{x}$, posterior mean error, and posterior variance.

$p_\phi(x|y,z,\mathscr{A}) = \mathscr{N}(x|x^K(z),\beta I)$. The latter is uniform across the pixels, and acts as a background. Thus, we show the HPDs of PIN with full variance (unbiased variance estimated by $\widehat{\mathbf{Cov}}_{p(x|y)}[x]$) and the variance without β factor (i.e., $\widehat{\mathbf{Cov}}_{p(x|y)}[x] - \beta I$) contains more structures in the credible intervals. Further, the overall shape and magnitude of the posterior credible intervals by PIN with the full variance and GM3 are fairly closely to each other, but there is notable difference in the cold regions (i.e. zero count): While GM3 still recovers non-zero variances in cold regions, PIN could provide almost zero variance, upon subtracting the background variance. In addition, posterior credible intervals of PIN without background variance can indicate the contrast of variance to highlight the pixels where variances of GM3 are also relatively higher. The comparison between the cross-section plots for low and moderate count cases (i.e., high and low noise levels, respectively) on the same ground truth phantom indicates that PIN does provide higher uncertainty for higher noise level, which is intuitively consistent with the underlying statistical background. However, one should interpret the differences of results with some caution: one common issue with UQ methods is the calibration of the uncertainty, and most deep learning based methods do not provide out-of-box

calibrated probabilities [84]. For large-scale inverse problems, e.g., PET, it is nontrivial to calibrate the uncertainty provided by UQ methods within the setting of implicit prior and likelihood function. Thus, one should not immediately conclude that the uncertainty recovered by PIN is less accurate than that GM3 or vice versa.

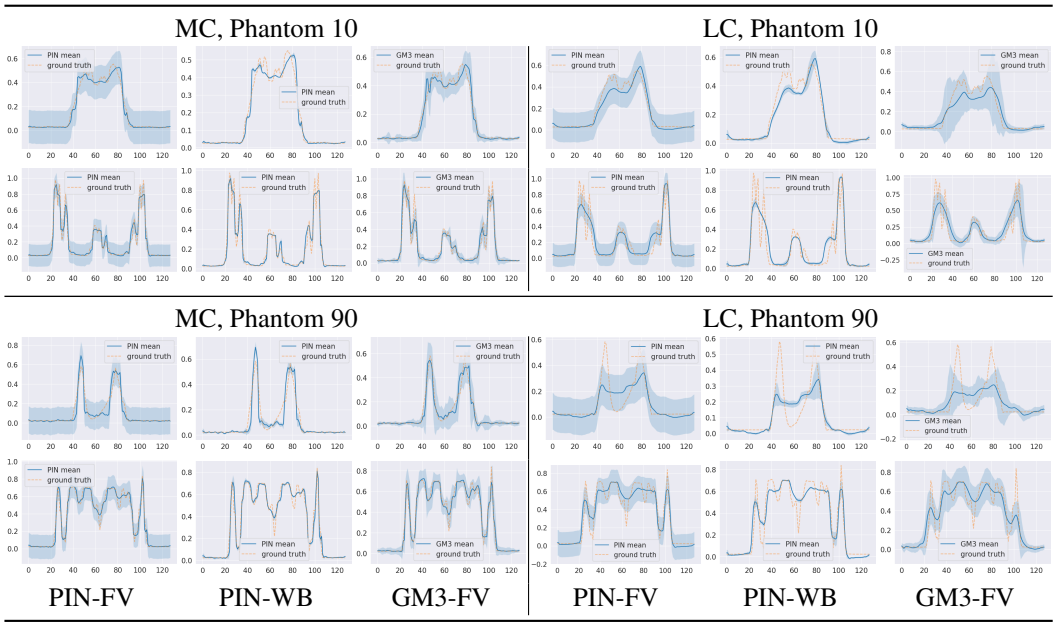


Figure 4.9: Comparison between PIN with full variance (PIN-FV), PIN without background variance (PIN-WB) and GM3 with full variance (GM3-FV), for BrainWeb phantoms 10 and 90 (size: 128×128) with the two peak values 1e4 (MC) and 1e2 (LC). Within each block, from left to right: sample mean and 0.95 credible interval of the 11th (top) and 101-th (bottom) horizontal slice.

4.5 Conclusion

In this work, we have developed a general and flexible probabilistic computational framework, termed as Probabilistic Iterative Networks, for uncertainty quantification of inverse problems in a purely data-driven setting. The approach is based on the conditional variational autoencoder loss, and employs the iterative deep neural network to recurrently refine the samples using the observation and forward map, seeded by a probabilistic encoder conditioned on the observation. The efficiency of the framework is underpined by encoding the observations in a low-dimensional latent space, and the flexibility (with datasets and problem dimensions) is facilitated by allowing explicitly inputting forward maps and their adjoints. The potentials of the framework have been demonstrated on PET image reconstruction with both moderate and low count levels, and the approach shows competitive performance when compared with several deterministic and probabilistic benchmark methods.

There are several avenues for further study. First, the framework is flexible and general, and it is of interest to apply it to other imaging modalities, e.g., MRI, CT and PET-MRI, especially in the undersampling / low-dose regime, for which there is a great demand on uncertainty quantification due to lack of information. Such studies will also shed insights into statistical features of the framework.

Second, theoretically, it is of much interest to analyse the asymptotic of the CVAE loss as an upper bound of the expected KL divergence. This line of research has been long-standing in variational inference, and often provides theoretical guarantees of the overall inference procedure and guidelines for constructing efficient approximations. Third, it is imperative to develop scalable benchmarks for uncertainty quantification of high-dimensional inverse problems. Several deep learning based uncertainty quantifications have been proposed in the machine learning literature, but mostly on different types of uncertainties or without explicitly elucidating the sources of uncertainties. Further, in this work we showed the flexibility of the framework by training on synthetic data and testing on more realistic data. Since the prior information is implicitly encoded into the framework from training data, the information captured by the framework from training data is not explicitly acquirable. As a result, it would be of great interest to further analyse the boundary of the information captured by one training. One possible way to investigate it is to track the performance decay of point estimate and uncertainty calibration by changing features in the test data in a variable controlling manner. However, how to measure the calibration of such high dimensional imaging problems with hidden prior information is also an open problem.

Chapter 5

Conclusions

Inverse problems with Poisson data represent an important class of real world problems and uncertainty quantification of possible unobservables for a given observation is vital to downstream decision making. However, current approaches mainly focus on point estimates and could not recover the information of uncertainty. While the Bayesian framework could provide a systematic scheme to solve this problem, full Bayesian approaches suffer from high computational complexities due to the curse of dimensionality and thus not scalable for real world problems, e.g., medical imaging.

In this thesis, we reviewed Bayesian inference techniques from the machine learning community and discussed their potentials for inverse problems with Poisson data. From the literature review, we concluded that 1) due to the special characteristics emerging in inverse problems with Poisson data, vanilla applications of these techniques to our problems is not straightforward and 2) neglecting the special characteristics would lead to inaccurate and inefficient practice of full Bayesian treatments. Aligning with this spirit, we further stressed three distinguishing characteristics, which are respectively one of the main points addressed in the three main chapters of this thesis.

In the first main chapter, i.e., Chapter 2, we studied variational inference with Gaussian approximations for Poisson data with exponential inverse link function. The studied model is a simplified version of that of X-ray CT and also referred to as Poisson regression in the statistics literature. We addressed the low rank structure of forward operators in this work and leveraged it to reduce the computational complexity of the inference algorithm. Besides the scalability of the algorithm, we also investigated several theoretical aspects, i.e., existence and uniqueness of the optimal Gaussian approximation, convergence properties of the algorithms, etc, which are largely missing in the literature. And we supported discussed theoretical and algorithmic properties by numerical experiments.

In the second main chapter, i.e., Chapter 3, we studied expectation propagation with Gaussian approximation for Poisson data with linear inverse link function, which is related to the model in emission tomography. In this work, we addressed how to incorporate the important nonnegativity property emerging in real world applications into the inference procedure. We showed detailed derivation of the algorithms, complexity and schemes for stable implementation for the case of a

Laplace type prior. The scalability is addressed by leveraging the rank 1 projection form in the factors of the posterior distribution for high dimensional moment evaluations. We showed the fast convergence, state-of-the-art results and capability to handle two dimensional PET images.

In the last main chapter, i.e., Chapter 4, we built a general and flexible probabilistic framework, i.e., Probabilistic Iterative Networks. This framework could be used for uncertainty quantification of inverse problems in a purely data-driven setting and is not limited to Poisson data. We addressed the incorporation of forward and adjoint operators into deep learning based approaches and achieved extra scalability of training due to the flexibility of dimensions. Besides, we also addressed the scalability by encoding the observations in a low-dimensional latent space, which could avoid direct sampling in a high dimensional space. We demonstrated potentials of the framework on PET image reconstructions and competitiveness by comparing with several deterministic and probabilistic benchmark methods.

To conclude, in this thesis, we studied bespoke Bayesian inference for inverse problems with Poisson data by taking into consideration vital and specific characteristics emerging in real world problems. While we conducted the research with the main focus on uncertainty quantification with scalability, we also investigated many other important aspects, e.g., theoretical understandings, practical concerns, framework generality. Following the same spectrum of perspectives, we now discuss several possible avenues for further study:

- The first and one of the most fundamental issues is the quality of the approximate posterior distributions relative to the true posterior distribution. Although approximate inference methods have achieved good accuracy on many practical problems, the lack of theoretical guarantees has been long standing in the literature.

- Second, even though the approximate inference methods could provide accurate approximations to the true posterior distribution, the quality of the posterior distribution itself may still suffer from the misspecification of the prior distribution and likelihood function. Frameworks theoretically and empirically studying the misspecification are needed, which could justify the calibration of the true posterior distribution.

- Third, many interesting links could be observed between approximate inference and classical regularisation theory. Especially in the lower bound functionals, the KL term between prior and approximate posterior acts as an analogue of Tikhonov regulariser. It is thus interesting to study consistency and convergence rates from the perspective of classical regularisation theory.

- Fourth, apart from the theoretical analysis of approximation quality, it is also imperative to develop scalable benchmarks to assess the uncertainty provided by approximate distributions for high-dimensional inverse problems. While one could employ Monte Carlo methods for

low dimensional problems, for high dimensional problems, especially in a data-driven setting where the forward model and prior distribution are implicit, how to build scalable and accurate benchmarks is highly non-trivial.

- We shed a light in terms of theoretical analysis in the first work, incorporate constraints in the second work and developed a general framework in the last work. It is very interesting to investigate how to benefit research of other similar problems in the field. For example, it is interesting to apply PIN to real world imaging problems, e.g., PET, MRI, CT and their synergies, especially scenarios where there is a great demand on uncertainty quantification due to lack of information.

- Moreover, although the full distributions (in an approximated way) are provided with full Bayesian approaches, it is still an open problem how to make best use of the uncertainty information. While one promising application is to conduct Bayesian hypothesis test based on the approximate distribution similar to [115], another interesting direction is to investigate how to leverage the uncertainty information to enhance downstream research, e.g., image segmentations of reconstructed images.

Appendix A

Appendix to Chapter 2

A.1 On the Iteration (2.12)

In this appendix, we discuss an interesting property of the iteration (2.12), for the initial guess $C^0 = C_0$. We denote the fixed point map in (2.12) by T, i.e.,

$$T(C) = (C_0^{-1} + A^t \operatorname{diag}(e^{A\bar{x}+\frac{1}{2}\operatorname{diag}(ACA^t)})A)^{-1}.$$

The next result gives the antimonotonicity of the map T on $\mathscr{S}_m^+$, i.e., for $C, \tilde{C} \in \mathscr{S}_m^+$, if $0 \leq C \leq \tilde{C}$, then $T(C) \geq T(\tilde{C})$.

Lemma A.1.1. *The mapping T is antimonotone.*

Proof. Let $C, \tilde{C} \in \mathscr{S}_m^+$. If $C \leq \tilde{C}$, then $\operatorname{diag}(ACA^t) \leq \operatorname{diag}(A\tilde{C}A^t)$ componentwise. The claim follows from the identity $T(C) - T(\tilde{C}) = T(C)A^t \operatorname{diag}(e^{A\bar{x}+\frac{1}{2}\operatorname{diag}(A\tilde{C}A^t)} - e^{A\bar{x}+\frac{1}{2}\operatorname{diag}(ACA^t)})AT(\tilde{C}) \geq 0$. □

The next result shows the monotonicity of the sequence $\{C^k\}$ generated by (2.12).

Lemma A.1.2. *For any initial guess $C^0 \in \mathscr{S}_m^+$, the sequence $\{C^k\}_{k\geq 0}$ generated by the iteration* (2.12) *has the following properties: (i) $C^k \geq 0$ for all $k \geq 0$; (ii) $C^k \leq C_0$ for all $k \geq 0$; (iii) If $C^k \geq C^j$ then $C^{k+1} \leq C^{j+1}$; (iv) If $C^k \geq C^j$ then $C^{k+2} \geq C^{j+2}$.*

Proof. Properties (i) and (ii) are obvious. Properties (iii) and (iv) are direct consequences of the fact that the map T is antimonotone on $\mathscr{S}_m^+$, cf. Lemma A.1.1. □

The next result shows that the sequence constitutes two subsequences, each converging to a fixed point of T^2, which implies either a periodic orbit of period 2 of the map T or a fixed point of T,

Theorem A.1.1. *With the initial guess $C^0 = C_0$, the sequence $\{C^k\}_{k\geq 0}$ generated by iteration* (2.12) *converges to a fixed-point of T^2.*

Proof. Lemma A.1.2(ii) implies

$$C^2 \leq C^0, \tag{A.1}$$

so we can use Lemma A.1.2(iv) inductively to argue that $\{C^{2k}\}_{k\geq 0}$ is a decreasing sequence. From (A.1) and Lemma A.1.2(iii), we deduce $C^1 \leq C^3$, which together with Lemma A.1.2(iv) implies that the sequence $\{C^{2k+1}\}_{k\geq 0}$ is increasing. By the boundedness and monotonicity, both $\{C^{2k}\}_{k\geq 0}$ and $\{C^{2k+1}\}_{k\geq 0}$ converge, with the limit C^* and C^{**}, respectively. These are the limits of the fixed point map T^2. □

Remark A.1.1. *By Lemma A.1.2, $C^* \geq C^{**}$, and if $C^* = C^{**}$, the whole sequence converges. Generally, the interval of matrices $[C^{**}, C^*]$ provides a lower and sharp bounds for the fixed point of the iteration* (2.12) *(which is a priori known to be unique and to exist). By repeating the argument in [41, Theorem 2.2], one may also examine the convergence of the sequence for the initial guess either $C^0 < C^{**}$ or $C^0 > C^*$.*

A.2 Differentiability of the Regularised Solution

In this part, we discuss the differentiability of the regularized solution $(\bar{x}_\alpha, C_\alpha)$ in α. For simplicity, we omit the subscript α. By differentiating (2.7) in α and chain rule, we obtain (with $\dot{\bar{x}} = \frac{d\bar{x}}{d\alpha}$ and $\dot{C} = \frac{dC}{d\alpha}$):

$$\begin{aligned} (A^t DA + \alpha C_0^{-1})\dot{\bar{x}} + \tfrac{1}{2}A^t D\mathrm{diag}(A\dot{C}A^t) &= -\bar{C}_0^{-1}(\bar{x} - \mu_0), \\ (C^{-1}\dot{C}C^{-1} + \tfrac{1}{2}A^t D^{\frac{1}{2}}\mathrm{diagdiag}(A\dot{C}A^t)D^{\frac{1}{2}}A) + A^t D^{\frac{1}{2}}\mathrm{diag}(A\dot{\bar{x}})D^{\frac{1}{2}}A &= -\bar{C}_0^{-1}, \end{aligned} \tag{A.2}$$

where $D = \mathrm{diag}(e^{A\bar{x} + \frac{1}{2}\mathrm{diag}(ACA^t)}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix. This constitutes a coupled linear system for $(\dot{\bar{x}}, \dot{C})$. The next result gives its unique solvability.

Theorem A.2.1. *The sensitivity system* (A.2) *is uniquely solvable.*

Proof. Since the system (A.2) is linear and square, it suffices to show that the homogeneous problem has only a zero solution. To this end, by eliminating the variable $\dot{\bar{x}}$ from the second line in (A.2) using the first line, we obtain the Schur complement for $\dot{C}$:

$$C^{-1}\dot{C}C^{-1} + \tfrac{1}{2}A^t D^{\frac{1}{2}}\mathrm{diagdiag}(A\dot{C}A^t)D^{\frac{1}{2}}A - \tfrac{1}{2}A^t D^{\frac{1}{2}}\mathrm{diag}(A(A^t DA + \alpha\bar{C}_0^{-1})^{-1}A^t D\mathrm{diag}(A\dot{C}A^t))D^{\frac{1}{2}}A.$$

For any fixed C, this defines a linear map on $\mathbb{R}^{m\times m}$. Next we show its invertibility. To this end, we take inner product the map with $\dot{C}$, and show its positivity. Clearly, the first term is strictly positive. Thus it suffices to consider the last two terms. By the cyclic property of trace, with $d = \mathrm{diag}(D) \in \mathbb{R}^n$, we have

$$\begin{aligned} &(A^t\mathrm{diag}(d \circ \mathrm{diag}(A\dot{C}A^t))A, \dot{C}) = \mathrm{tr}(A^t\mathrm{diag}(d \circ \mathrm{diag}(A\dot{C}A^t))A\dot{C}) \\ =&(D\mathrm{diag}(\mathrm{diag}(A\dot{C}A^t)), A\dot{C}A^t) = (D\mathrm{diag}(A\dot{C}A^t), \mathrm{diag}(A\dot{C}A^t)) = (\bar{e}, \bar{e}), \end{aligned}$$

where $\bar{e} = D^{\frac{1}{2}}\mathrm{diag}(A\dot{C}A^t) \in \mathbb{R}^n$. Similarly, by letting $\bar{A} = D^{\frac{1}{2}}A$, we have

$$\begin{aligned}&(A^tD\mathrm{diag}(A(A^tDA+\alpha\bar{C}_0^{-1})^{-1}A^tD\mathrm{diag}(A\dot{C}A^t))A,\dot{C})\\ =&(D\mathrm{diag}(A(A^tDA+\alpha\bar{C}_0^{-1})^{-1}A^tD\mathrm{diag}(A\dot{C}A^t)),A\dot{C}A^t)\\ =&(\bar{A}(\bar{A}^t\bar{A}+\alpha\bar{C}_0^{-1})^{-1}\bar{A}^t\bar{e},\bar{e}).\end{aligned}$$

Since $I_n-\bar{A}(\bar{A}^t\bar{A}+\alpha\bar{C}_0^{-1})^{-1}\bar{A}^t>0$, the associated bilinear form is coercive on $\mathscr{S}_m^+$. Thus the Schur complement is invertible, and the system (A.2) has a unique solution. □

Corollary A.2.1. *For any rank deficient A, $\dot{C}\neq 0$.*

Proof. If $\dot{C}=0$, the second equation in (A.2) reduces to $A^tD^{\frac{1}{2}}\mathrm{diag}(A\dot{x})D^{\frac{1}{2}}A=-\bar{C}_0^{-1}$. By assumption, A is rank deficient, and thus the left hand side is rank deficient, whereas the right hand side is of full rank, which leads to a contradiction. Thus we have $\dot{C}\neq 0$. □

The next result gives a lower-bound for the derivative $\frac{d}{d\alpha}\psi(\bar{x}_\alpha,C_\alpha)$.

Theorem A.2.2. *The functional $\psi(\bar{x}_\alpha,C_\alpha)$ satisfies*

$$\frac{d}{d\alpha}\psi(\bar{x}_\alpha,C_\alpha)\geq\alpha(C_0^{-1}\dot{\bar{x}},\dot{\bar{x}})+\tfrac{1}{2}(C^{-1}\dot{C}C^{-1},\dot{C}).$$

Proof. By the definition of the functional ψ, we have

$$\frac{d}{d\alpha}\psi(\bar{x}_\alpha,C_\alpha)=-(\bar{C}_0^{-1}(\bar{x}-\mu_0),\dot{\bar{x}})-\tfrac{1}{2}(\bar{C}_0^{-1},\dot{C}).$$

By taking inner product the first equation in (A.2) with $\dot{\bar{x}}$, and the second with $\frac{1}{2}\dot{C}$, we get

$$\begin{aligned}((A^tDA+\alpha C_0^{-1})\dot{\bar{x}},\dot{\bar{x}})+\tfrac{1}{2}(A^tD\mathrm{diag}(A\dot{C}A^t),\dot{\bar{x}})&=-(\bar{C}_0^{-1}(\bar{x}-\mu_0),\dot{\bar{x}}),\\ \tfrac{1}{2}(C^{-1}\dot{C}C^{-1}+\tfrac{1}{2}A^tD^{\frac{1}{2}}\mathrm{diagdiag}(A\dot{C}A^t)D^{\frac{1}{2}}A,\dot{C})+\tfrac{1}{2}(A^tD^{\frac{1}{2}}\mathrm{diag}(A\dot{\bar{x}})D^{\frac{1}{2}}A,\dot{C})&=-\tfrac{1}{2}(\bar{C}_0^{-1},\dot{C}).\end{aligned}$$

By the cyclic property of trace and summing these two identities, we obtain

$$\begin{aligned}-(\bar{C}_0^{-1}(\bar{x}-\mu_0),\dot{\bar{x}})-\tfrac{1}{2}(\bar{C}_0^{-1},\dot{C})=&(\alpha C_0^{-1}\dot{\bar{x}},\dot{\bar{x}})+\tfrac{1}{2}(C^{-1}\dot{C}C^{-1},\dot{C})\\ &+(A^tDA\dot{\bar{x}},\dot{\bar{x}})+\tfrac{1}{4}(D\mathrm{diag}(A\dot{C}A^t),\mathrm{diag}(A\dot{C}A^t))\\ &+(D^{\frac{1}{2}}\mathrm{diag}(A\dot{C}A^t),D^{\frac{1}{2}}A\dot{\bar{x}}).\end{aligned}\tag{A.3}$$

Meanwhile, by the Cauchy-Schwarz inequality, we have

$$(D^{\frac{1}{2}}\mathrm{diag}(A\dot{C}A^t),D^{\frac{1}{2}}A\dot{\bar{x}})\geq-(D^{\frac{1}{2}}A\dot{\bar{x}},D^{\frac{1}{2}}A\dot{\bar{x}})-\tfrac{1}{4}(D^{\frac{1}{2}}\mathrm{diag}(A\dot{C}A^t),D^{\frac{1}{2}}\mathrm{diag}(A\dot{C}A^t)).$$

Substituting the preceding inequality into (A.3) yields the desired estimate. □

Corollary A.2.2. *The functional $\psi(\bar{x}_\alpha, C_\alpha)$ is strictly increasing in α.*

Proof. By Theorem A.2.1, (A.2) is uniquely solvable. Since the right hand side of (A.2) is nonvanishing (by assumption, C_0 is nonzero), the solution pair $(\dot{\bar{x}}, \dot{C})$ to (A.2) is nonzero. Thus, by Theorem A.2.2, $\frac{d}{d\alpha}\psi(\bar{x}_\alpha, C_\alpha)$ is strictly positive, i.e., $\psi(\bar{x}_\alpha, C_\alpha)$ is strictly increasing. □

Remark A.2.1. *For the standard regularised least-squares problem, the solution is distinct for different α, and it never vanishes (except the trivial case $y = 0$). The proof in Corollary A.2.2 indicates that an analogous statement holds for the Poisson model* (2.2).

Appendix B

Appendix to Chapter 3

B.1 Parameterising Gaussian Distributions

For a Gaussian distribution $\mathcal{N}(x|\mu,C)$ with mean $\mu \in \mathbb{R}^n$ and covariance $C \in \mathcal{S}_+^n$, the probability density $\pi(x|\mu,C)$ is given by

$$\pi(x|\mu,C) = (2\pi)^{-\frac{n}{2}}|C|^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\mu)^t C^{-1}(x-\mu)} = e^{\zeta + h^t x - \frac{1}{2}x^t \Lambda x},$$

where the parameters $\Lambda \in \mathcal{S}_+^n$, $h \in \mathbb{R}^n$ and $\zeta \in \mathbb{R}$ are respectively given by

$$\Lambda = C^{-1}, \quad h = \Lambda\mu, \quad \text{and} \quad \zeta = -\tfrac{1}{2}(n\log 2\pi + \log|\Lambda| + \mu^t\Lambda\mu).$$

Thus, the density function $\pi(x|\mu,C)$ is also uniquely defined by Λ and h. In the literature, Λ is often referred to as the precision matrix and h as the precision mean. And the pair (h,Λ) is called the natural parameter of a Gaussian distribution.

It is easy to check that the product of k Gaussians $\{\mathcal{N}(x|\mu_k,C_k)\}_{k=1}^m$ is also a Gaussian $\mathcal{N}(x|\mu,C)$ after normalisation, and the mean μ and covariance C of the product are given by

$$\mu = C\sum_{k=1}^m C_k^{-1}\mu_k \quad \text{and} \quad C = \Big(\sum_{k=1}^m C_k^{-1}\Big)^{-1}, \tag{B.1}$$

or equivalently

$$h = \sum_{k=1}^m h_k \quad \text{and} \quad \Lambda = \sum_{k=1}^m \Lambda_k. \tag{B.2}$$

Appendix C

Appendix to Chapter 4

C.1 Proof of Proposition 4.3.1

In this section, we show the proof of Proposition 4.3.1 following repeating it.

Proposition C.1.1. *Optimising $\mathscr{L}_{\mathrm{CVAE}}(\theta,\phi;x,y)$ (expected on the training data distribution) is equivalent to optimising an upper bound of the expected reversed KL divergence*

$$J^*(p(x|y)) = \mathbb{E}_{p^*(y)}[D_{\mathrm{KL}}(p^*(x|y)||p(x|y))].$$

Proof. By the definition of KL divergence and Fubini theorem,

$$\begin{aligned} J^*(p(x|y)) &= \mathbb{E}_{p^*(y)}\mathrm{KL}(p^*(x|y)||p(x|y)) \\ &= \int p^*(y)\int p^*(x|y)\log\frac{p^*(x|y)}{p(x|y)}\mathrm{d}x\mathrm{d}y \\ &= \int p^*(x,y)[\log p^*(x|y) - \log p(x|y)]\mathrm{d}(x,y) \\ &= \mathbb{E}_{p^*(x,y)}[\log p^*(x|y)] + \mathbb{E}_{p^*(x,y)}[-\log p(x|y)]. \end{aligned} \tag{C.1}$$

Then by the classical derivation of a lower bound for the logarithm $\log p(x|y)$ of the conditional

distribution $p(x|y)$,

$$\begin{aligned}
\log p(x|y) &= \int q(z|x,y)\log p(x|y)\mathrm{d}z \\
&= \int q(z|x,y)\log\frac{p(x|y)p(z|x,y)}{p(z|x,y)}\mathrm{d}z \\
&= \int q(z|x,y)\log\frac{p(x,z|y)}{p(z|x,y)}\mathrm{d}z \\
&= \int q(z|x,y)\log\frac{p(x,z|y)}{q(z|x,y)}\frac{q(z|x,y)}{p(z|x,y)}\mathrm{d}z \\
&= \int q(z|x,y)\log\frac{p(x,z|y)}{q(z|x,y)}\mathrm{d}z + \int q(z|x,y)\log\frac{q(z|x,y)}{p(z|x,y)}\mathrm{d}z \\
&\geq \int q(z|x,y)\log\frac{p(x,z|y)}{q(z|x,y)}\mathrm{d}z \\
&= \int q(z|x,y)\log\frac{p(z|y)p(x|y,z)}{q(z|x,y)}\mathrm{d}z \\
&= \int q(z|x,y)\log\frac{p(z|y)}{q(z|x,y)}\mathrm{d}z + \int q(z|x,y)\log p(x|y,z)\mathrm{d}z \\
&= -\mathrm{KL}(q(z|x,y)||p(z|y)) + \mathbb{E}_{z\sim q(z|x,y)}[\log p(x|y,z)],
\end{aligned}$$

where the inequality is due to the nonnegativity of the Kullback-Leibler divergence. Consequently, we have

$$-\log p(x|y) \leq \mathrm{KL}(q(z|x,y)||p(z|y)) + \mathbb{E}_{z\sim q(z|x,y)}[-\log p(x|y,z)]. \tag{C.2}$$

Substituting inequity (C.2) into equation (C.1) yields

$$J^*(p(x|y)) \leq \mathbb{E}_{p^*(x,y)}[\log p^*(x|y)] + J.$$

Since the term $\mathbb{E}_{p^*(x,y)}[\log p^*(x|y)]$ is independent of the variational distribution $p(x|y)$ and other auxiliary distributions introduced with z, minimising J is equivalent to minimising an upper bound of $J^*(p(x|y))$. □

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Courier Corporation, 1965.

[3] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[4] J. Adler and O. Öktem. Deep bayesian inversion. *arXiv preprint arXiv:1811.05910*, 2018.

[5] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 131–142, 1966.

[6] S.-I. Amari. Differential geometry of curved exponential families-curvatures and information loss. *The Annals of Statistics*, pages 357–385, 1982.

[7] S.-I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[8] S.-i. Amari. *Differential-geometrical methods in statistics*, volume 28. Springer Science & Business Media, 2012.

[9] W. N. Anderson, Jr., G. B. Kleindorfer, P. R. Kleindorfer, and M. B. Woodroofe. Consistent estimates of the parameters of a linear system. *Ann. Math. Stat.*, 40:2064–2075, 1969.

[10] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

[11] C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor. Gaussian process approximations of stochastic differential equations. *JMLR: Workshop Conf. Proc.*, 1:1–16, 2007.

[12] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223, 2017.

[13] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numer.*, 28:1–174, 2019.

[14] S. R. Arridge, K. Ito, B. Jin, and C. Zhang. Variational Gaussian approximation for Poisson data. *Inverse Problems*, 34(2):025005, 29 pp., 2018.

[15] D. Barber and C. M. Bishop. Ensemble learning in Bayesian neural networks. *NATO ASI Series F Comput. Syst. Sci.*, 168:215–238, 1998.

[16] J. M. Bardsley and A. Luttman. A Metropolis-Hastings method for linear inverse problems with Poisson likelihood and Gaussian prior. *Int. J. Uncertain. Quantif.*, 6(1):35–55, 2016.

[17] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini. Image deblurring with poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006, 2009.

[18] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, second edition, 1999.

[19] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Singapore, 2006.

[20] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: a review for statisticians. *J. Amer. Statist. Assoc.*, 112(518):859–877, 2017.

[21] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622, 2015.

[22] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[23] T. Bui, D. Hernández-Lobato, J. Hernandez-Lobato, Y. Li, and R. Turner. Deep gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.

[24] X. Cai, M. Pereyra, and J. D. McEwen. Uncertainty quantification for radio interferometric imaging: Ii. map estimation. *Monthly Notices of the Royal Astronomical Society*, 480(3):4170–4182, 2018.

[25] A. Cameron and P. K. Trivedi. *Regression Analysis of Count Data*. Cambridge University Press, 1998.

[26] E. Challis and D. Barber. Gaussian kullback-leibler approximate inference. *The Journal of Machine Learning Research*, 14(1):2239–2286, 2013.

[27] T. F. Chan and J. Shen. *Image Processing and Analysis*. SIAM, Philadelphia, PA, 2005.

[28] H. Chen, Y. Zhang, Y. Chen, J. Zhang, W. Zhang, H. Sun, Y. Lv, P. Liao, J. Zhou, and G. Wang. LEARN: learned experts' assessment-based reconstruction network for sparse-data CT. *IEEE Trans. Med. Imag.*, 37(6):1333–1347, 2018.

[29] Y. Chen, M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, M. Botvinick, and N. de Freitas. Learning to learn without gradient descent by gradient descent. In *Proceedings of the 34th International Conference on Machine Learning*, pages 748–756, 2017.

[30] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike, and A. C. Evans. Brainweb: Online interface to a 3d MRI simulated brain database. *NeuroImage*, 5(4):part 2/4, S425, 1997.

[31] I. Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108, 1964.

[32] J. P. Cunningham, P. Hennig, and S. Lacoste-Julien. Gaussian probabilities and expectation propagation. Preprint, arXiv:1111.6832, 2011.

[33] A. R. De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE transactions on medical imaging*, 14(1):132–137, 1995.

[34] G. Dehaene and S. Barthelmé. Expectation propagation in the large data limit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):199–217, 2018.

[35] G. P. Dehaene and S. Barthelmé. Bounding errors of expectation-propagation. In *Advances in Neural Information Processing Systems*, pages 244–252, 2015.

[36] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous. Tensorflow distributions. Preprint, arXiv:1711.10604, 2017.

[37] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV 2014: Computer Vision*, pages 184–199, 2014.

[38] A. Durmus, E. Moulines, and M. Pereyra. Efficient bayesian computation by proximal markov chain monte carlo: when langevin meets moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018.

[39] P. S. Dwyer. Some applications of matrix derivatives in multivariate analysis. *J. Amer. Stat. Assoc.*, 62(318):607–625, 1967.

[40] M. J. Ehrhardt, K. Thielemans, L. Pizarro, D. Atkinson, S. Ourselin, B. F. Hutton, and S. R. Arridge. Joint reconstruction of PET-MRI by exploiting structural similarity. *Inverse Problems*, 31(1):015001, 23 pp., 2015.

[41] S. M. El-Sayed and A. C. M. Ran. On an iteration method for solving a class of nonlinear matrix equations. *SIAM J. Matrix Anal. Appl.*, 23(3):632–645, 2001/02.

[42] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic, Dordrecht, 1996.

[43] H. Erdoğan and J. A. Fessler. Monotonic algorithms for transmission tomography. *IEEE Trans. Med. Imag.*, 18(9):801–814, 1999.

[44] J. A. Fessler and H. Erdogan. A paraboloidal surrogates algorithm for convergent penalized-likelihood emission image reconstruction. In *Nuclear Science Symposium, 1998. Conference Record. 1998 IEEE*, volume 2, pages 1132–1135. IEEE, 1998.

[45] J. A. Fessler and A. O. Hero. Penalized maximum-likelihood image reconstruction using space-alternating generalized em algorithms. *IEEE Transactions on Image Processing*, 4(10):1417–1429, 1995.

[46] C. Finn. *Learning to Learn with Gradients*. PhD thesis, UC Berkeley, 2018.

[47] I. M. Franck and P. S. Koutsourelakis. Sparse variational Bayesian approximations for nonlinear inverse problems: applications in nonlinear elastography. *Comput. Methods Appl. Mech. Engrg.*, 299:215–244, 2016.

[48] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[49] B. Gärtner and J. Matousek. *Approximation Algorithms and Semidefinite Programming*. Springer-Verlag, Berlin, Heidelberg,, 2012.

[50] J. Gast and S. Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3369–3378, 2018.

[51] M. Gehre. *Rapid Uncertainty Quantification for Nonlinear Inverse Problems*. PhD thesis, University of Bremen, Bremen, 2013.

[52] M. Gehre and B. Jin. Expectation propagation for nonlinear inverse problems–with an application to electrical impedance tomography. *Journal of Computational Physics*, 259:513–535, 2014.

[53] A. Gelman, A. Vehtari, P. Jylänki, C. Robert, N. Chopin, and J. P. Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*, 2014.

[54] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[55] A. Graves. Practical variational inference for neural networks. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2348–2356, 2011.

[56] P. J. Green, K. Łatuszyński, M. Pereyra, and C. P. Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Statistics and Computing*, 25(4):835–862, 2015.

[57] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[58] D. Hafner, D. Tran, A. Irpan, T. Lillicrap, and J. Davidson. Reliable uncertainty estimates in deep neural networks using noise contrastive priors. *arXiv preprint arXiv:1807.09289*, 2018.

[59] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[60] P. Hall, J. T. Ormerod, and M. P. Wand. Theory of Gaussian variational approximation for a Poisson mixed model. *Stat. Sinica*, 21(1):369–389, 2011.

[61] J. M. Hernández-Lobato, Y. Li, M. Rowland, D. Hernández-Lobato, T. Bui, and R. E. Turner. Black-box α-divergence minimization. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1511–1520, 2016.

[62] G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT'93, Proc. 6th Annual Conf. Comput. Learning Theory*, pages 5–13, New York, 1993. ACM.

[63] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[64] T. Hohage and F. Werner. Inverse problems with Poisson data: statistical regularization theory, applications and algorithms. *Inverse Problems*, 32(9):093001, 56, 2016.

[65] A. Horé and D. Ziou. Image quality metrics: PSNR vs. SSIM. In *20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[66] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo. Deep learning for undersampled MRI reconstruction. *Phys. Med. Biol.*, 63(13):135007, 15 pp., 2018.

[67] K. Ito and B. Jin. *Inverse Problems: Tikhonov Theory and Algorithms*. World Scientific, Hackensack, NJ, 2015.

[68] K. Ito, B. Jin, and T. Takeuchi. A regularization parameter for nonsmooth Tikhonov regularization. *SIAM J. Sci. Comput.*, 33(3):1415–1438, 2011.

[69] W. James and C. Stein. Estimation with quadratic loss. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, pages 361–379. Univ. California Press, Berkeley, Calif., 1961.

[70] B. Jin. A variational Bayesian method to inverse problems with impulsive noise. *J. Comput. Phys.*, 231(2):423–435, 2012.

[71] B. Jin and J. Zou. Augmented Tikhonov regularization. *Inverse Problems*, 25(2):025001, 25, 2009.

[72] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Mach. Learning*, 37(2):183–233, 1999.

[73] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer-Verlag, New York, 2005.

[74] E. Kang, J. Min, and J. C. Ye. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med. Phys.*, 44(10):e360–e375, 2017.

[75] L. Kaufman. Implementing and accelerating the EM algorithm for positron emission tomography. *IEEE Trans. Med. Imag.*, 6(1):37–51, 1987.

[76] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, PA, 1995.

[77] A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017.

[78] E. Khan, S. Mohamed, and K. P. Murphy. Fast bayesian inference for non-conjugate gaussian process regression. In *Advances in Neural Information Processing Systems*, pages 3140–3148, 2012.

[79] M. E. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*, 2018.

[80] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*, 2014.

[81] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

[82] Y.-J. Ko and M. W. Seeger. Expectation propagation for rectified linear poisson regression. In G. Holmes and T.-Y. Liu, editors, *Asian Conference on Machine Learning*, volume PMLR 45, pages 253–268, 2016.

[83] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22:79–86, 1951.

[84] A. Kumar, P. S. Liang, and T. Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, pages 3787–3798, 2019.

[85] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[86] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[87] Y. Li, J. M. Hernández-Lobato, and R. E. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems*, pages 2323–2331, 2015.

[88] Y. Li and R. E. Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

[89] H. Lim, Y. K. Dewaraja, and J. A. Fessler. A pet reconstruction formulation that enforces non-negativity in projection space for bias reduction in y-90 imaging. *Physics in Medicine & Biology*, 63(3):035042, 2018.

[90] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York, 2001.

[91] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.

[92] Y. Lu, A. M. Stuart, and H. Weber. Gaussian approximations for probability measures on $\mathbf{R}^d$. SIAM/ASA J. Uncertainty Quantification, in press. arXiv:1611.08642, 2016.

[93] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Proc. Mag.*, 35(1):20–36, 2018.

[94] D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[95] A. Malinin and M. Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018.

[96] M. T. McCann, K. H. Jin, and M. Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Sig. Proc. Mag.*, 34(6):85–95, 2017.

[97] T. Minka. Power ep. Technical report, Technical report, Microsoft Research, Cambridge, 2004.

[98] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[99] T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[100] M. Moeller, T. Möllenhoff, and D. Cremers. Controlling neural networks via energy dissipation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[101] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte Carlo gradient estimation in machine learning. Prepring, arXiv:1906.10652, 2019.

[102] D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[103] J. T. Ormerod and M. P. Wand. Gaussian variational approximate inference for generalized linear mixed models. *J. Comput. Graph. Statist.*, 21(1):2–17, 2012.

[104] K. Osawa, S. Swaroop, M. E. E. Khan, A. Jain, R. Eschenhagen, R. E. Turner, and R. Yokota. Practical deep learning with bayesian principles. In *Advances in Neural Information Processing Systems 32*, pages 4289–4301. 2019.

[105] M. Pereyra. Maximum-a-posteriori estimation with bayesian confidence regions. *SIAM Journal on Imaging Sciences*, 10(1):285–302, 2017.

[106] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. O. Hero, and S. McLaughlin. A survey of stochastic simulation and optimization methods in signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):224–241, 2015.

[107] S. Petrone, J. Rousseau, and C. Scricciolo. Bayes and empirical bayes: do they merge? *Biometrika*, 101(2):285–302, 2014.

[108] D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *J. Assoc. Comput. Mach.*, 9:84–97, 1962.

[109] J. Pillow. Likelihood-based approaches to modeling the neural code. In K. Doya, S. Ishii, A. Pouget, and R. Rao, editors, *Bayesian Brain: Probabilistic Approaches to Neural Coding*, pages 53–70. MIT Press, Cambridge, 2007.

[110] F. J. Pinski, G. Simpson, A. M. Stuart, and H. Weber. Kullback-Leibler approximation for probability measures on infinite-dimensional spaces. *SIAM J. Math. Anal.*, 47(6):4091–4122, 2015.

[111] J. Qi and R. M. Leahy. Iterative reconstruction techniques in emission computed tomography. *Phys. Med. Biol.*, 51(15):R541–R578, 2006.

[112] C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

[113] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electron. J. Stat.*, 5:935–980, 2011.

[114] A. Rényi. On measures of entropy and information. Technical report, HUNGARIAN ACADEMY OF SCIENCES Budapest Hungary, 1961.

[115] A. Repetti, M. Pereyra, and Y. Wiaux. Scalable bayesian uncertainty quantification in imaging inverse problems via convex optimization. *SIAM Journal on Imaging Sciences*, 12(1):87–118, 2019.

[116] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, volume 32, pages 1278–1286, 2014.

[117] H. Robbins and S. Monro. A stochastic approximation method. In *Herbert Robbins Selected Papers*, pages 102–109. Springer, 1985.

[118] C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[119] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, second edition, 2004.

[120] D. Rohde and M. P. Wand. Semiparametric mean field variational Bayes: general principles and numerical issues. *J. Mach. Learn. Res.*, 17:Paper No. 172, 47, 2016.

[121] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[122] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Learning to deblur. *IEEE trans. Pattern Anal. Mach. Intel.*, 38(7):1439–1451, 2015.

[123] T. Schuster, B. Kaltenbacher, B. Hofmann, and K. S. Kazimierski. *Regularization Methods in Banach Spaces*. Walter de Gruyter GmbH & Co. KG, Berlin, 2012.

[124] M. W. Seeger. Bayesian inference and optimal design for the sparse linear model. *J. Mach. Learn. Res.*, 9:759–813, 2008.

[125] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imag.*, 1(2):113–122, 1982.

[126] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[127] S. Sotthivirat and J. A. Fessler. Image recovery using partitioned-separable paraboloidal surrogate coordinate ascent algorithms. *IEEE Trans. Imag. Proc.*, 11(3):306–317, 2002.

[128] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numer.*, 19:451–559, 2010.

[129] L. Tierney and J. B. Kadane. Accurate approximations for posterior moments and marginal densities. *J. Amer. Stat. Assoc.*, 81(393):82–86, 1986.

[130] K. Van Slambrouck, S. Stute, C. Comtat, M. Sibomana, F. H. van Velden, R. Boellaard, and J. Nuyts. Bias reduction for low-statistics PET: maximum likelihood reconstruction with a modified poisson distribution. *IEEE Trans. Med. Imag.*, 34(1):126–136, 2015.

[131] Y. Vardi, L. Shepp, and L. Kaufman. A statistical model for positron emission tomography. *J. Amer. Stat. Assoc.*, 80(389):8–20, 1985.

[132] L. Vargas, M. Pereyra, and K. C. Zygalakis. Accelerating proximal markov chain monte carlo by using explicit stabilised methods. *arXiv preprint arXiv:1908.08845*, 2019.

[133] M. Vono, N. Dobigeon, and P. Chainais. Bayesian image restoration under poisson noise and log-concave prior. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1712–1716. IEEE, 2019.

[134] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends©Mach. Learn.*, 1(1–2):1–305, 2008.

[135] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851, 2016.

[136] J. Wang and N. Zabaras. Hierarchical Bayesian models for inverse problems in heat conduction. *Inverse Problems*, 21(1):183–206, 2005.

[137] H. Xiang and J. Zou. Randomized algorithms for large-scale inverse problems with general tikhonov regularizations. *Inverse Problems*, 31(8):085008, 2015.

[138] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014.

[139] M. Yavuz and J. A. Fessler. New statistical models for randoms-precorrected PET scans. In *Information Processing in Medical Imaging (Lecture Notes in Computer Science)*, volume 1230, pages 190–203. Springer-Verlag, 1997.

[140] C. Zhang, S. Arridge, and B. Jin. Expectation propagation for poisson data. *Inverse Problems*, 35(8):085006, 2019.

[141] C. Zhang and B. Jin. Probabilistic residual learning for aleatoric uncertainty in image restoration. *arXiv preprint arXiv:1908.01010*, 2019.

[142] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Imag. Proc.*, 26(7):3142–3155, 2017.

[143] K. Zhang, W. Zuo, and L. Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Imag. Proc.*, 27(9):4608–4622, 2018.

[144] Y. Zhang and W. E. Leithead. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *J. Stat. Comput. Simul.*, 77(4):329–348, 2007.

[145] Q. Zhou, T. Yu, X. Zhang, and J. Li. Bayesian inference and uncertainty quantification for medical image reconstruction with poisson data. *SIAM Journal on Imaging Sciences*, 13(1):29–52, 2020.