Expectation Propagation

Valerie Bradley, Alan Chau, Anastasia Ignatieva

September 24, 2019

Abstract

Expectation Propagation (EP) is a popular method for variational inference in posterior studies. EP computes a product of exponential families aiming to minimise the Kullback-Leibler divergence between the target distribution and the approximations itself. In this project, we will explore into the theoretical property of EP based on the work of Dehaene and Barthelmé [2015] and Dehaene and Barthelmé [2018]. We implemented the EP algorithm described by Rasmussen and Williams [2006] and applied it to a toy example; we also used existing software to apply EP to a digit classification problem.

1 Introduction

Inference on posterior distributions in Bayesian statistics can be group into two main approaches: Markov chain Monte Carlo (MCMC) type sampling schemes and deterministic variational approximations. In contrast to MCMC, variational approximations are not guaranteed to return the true posterior, but provide fast deterministic approximations to arbitrary distributions. Variational methods are preferable to MCMC if computation time is a concern. Well-known examples of variational methods include the mean-field family, Laplace approximation and Expectation Propagation (EP), the main focus on this project.

Expectation Propagation [Minka, 2001] is a popular method for variational approximation in posterior analysis. It is an iterative algorithm to approximate complicated distributions, by finding a product of local (usually, though not always) Gaussian approximations.

Recall the density of a Gaussian distribution with mean μ and variance σ^2 can be expressed in the following form,

$$N(x|\mu,\sigma^2) \propto \exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\} \tag{1}$$

$$\propto \exp(-\frac{1}{2}\beta x^2 + rx) \tag{2}$$

where $\beta = \sigma^{-2}$, $r = \beta\mu$ are the natural parameters of the Gaussian exponential family. β is known as the precision and r is known as the linear shift. Expressing the distribution in terms of its natural parameters simplifies the computation of products of Gaussians.

Again, the goal of EP is to compute a Gaussian approximation $q(x)$ of a target distribution, $p(x) \propto \exp\{-\psi(x)\}$. In typical application of EP, $p(x)$ will be a posterior that can be factorized into n factor functions (also known as "sites" in the EP terminology):

$$p(x) = \prod_{i=1}^{n} l_i(x)$$

EP produces an approximation to the target with the same factor structure: n Gaussian factors $f_i(x)$ such that $q(x) = \prod_{i=1}^{n} f_i(x)$ and each factor $f_i(x)$ approximates the exact posterior at the corresponding sites $l_i(x)$. To produce such approximation, EP aims to solve:

$$\arg\min_{q\in\mathcal{Q}} KL(p||q) \tag{3}$$

where KL denotes the Kullback-Leibler divergence. Note that EP can be used with other exponential approximating families.

EP Algorithm

We will first describe the algorithm, and then provide an intuitive explanation. We initialise the algorithm by setting all $q_i = 1$. At iteration t, from a current approximation $q^t(x) = \prod_{i=1}^n f_i(x|r_i^t, \beta_i^t)$, we select a random site i to update. We then,

- Compute a *cavity* distribution $q_{-i}^t \propto \prod_{j \neq i} q_j^t(x)$ to remove the contribution of approximation from the i^{th} site. This can be done easily in terms of natural parameters:

$$q_{-i}(x) \propto \exp\Big(\big(\sum_{j\neq i} r_j^t\big)x - \big(\sum_{j \neq i} \beta_j^t\big)\frac{x^2}{2}\Big) \tag{4}$$

- Compute the *hybrid* distribution as the product of the *cavity* distribution and the exact likelihood at the i^{th} site $h_i^t(x) \propto q_{-i}^t(x) l_i(x)$
- Compute the Gaussian which minimizes the KL divergence to the hybrid,

$$\mathcal{P}(h_i^t) = \arg\min_q (KL(h_i^t||q)) \tag{5}$$

  In this case of Gaussian family, this is the same as matching the first two moments of h_i^t.

- Finally, update the approximation of f_i:

$$q_i^{t+1} = \frac{\mathcal{P}(h_i^t)}{q_{-i}^t} \tag{6}$$

  This can be done easily with closed form expressions as both numerator and denominator are Gaussian.

The above updating is applied to every site iteratively until convergence to a fixed point. EP is not deterministic in the sense that several fixed points can exist. However as stated in Dehaene and Barthelmé [2018], the mean and variance of all fixed points q are the same (though the individual approximations may differ). It is this fact that Deanene et.al utilised to derive tight bounds on the possible positions of these fixed points.

Intuition to EP

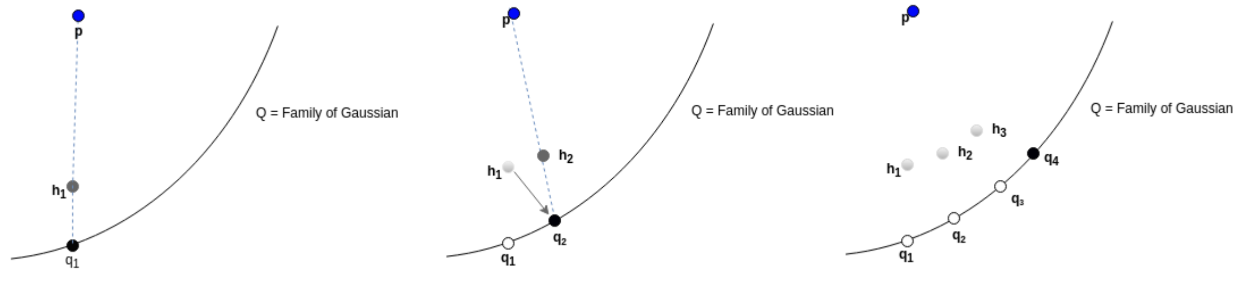

Figure 1: Conceptual illustration of how and why EP works, p denotes the target distribution, h_t and q_t denote the t^{th} run of the approximation. Based on Barthelmé [2016].

Figure 1 provides an illustration on how EP works in a conceptual framework [Barthelmé, 2016]. Imagine the curved line is the family of Gaussians – this will be the space where our approximations live. From the initial approximation q_1, in order to get "closer" to the target p, we create the hybrid h_1 distribution. This is then projected back to the family of Gaussians again and we repeat this process until convergence (Point q_4 in the figure).

EP is conceptually very simple and neat and has strong practical results to back up its success. In contrast, there exist only a few theoretical guarantees regarding its behavior. In the next section, we will study some theoretical results proposed by Dehaene and Barthelmé [2015] and Dehaene and Barthelmé [2018].

2 Theoretical Properties of EP

In this section we will discuss the performance of the EP approximations under both the finite [Dehaene and Barthelmé, 2018] and infinite data [Dehaene and Barthelmé, 2015] regimes. Both analyses require an assumption that the target distribution $p(x)$ is strongly-log-concave. This allows us to compute bounds on the distance between the mean and variance of the target distribution and of the approximation given by EP. We assume that all sites $l_i(x)$ are β_m-strongly log-concave with a slowly-changing log-function. That is, if we denote $\phi_i(x) = -\log(l_i(x))$ for all sites:

$$\forall i \forall x \quad \phi_i^{''}(x) \geq \beta_m > 0 \tag{7}$$

$$\forall i \forall d \in [3,4,5,6] \quad |\phi_i^{(d)}(x)| \leq K_d \tag{8}$$

Then naturally the target distribution will inherit these properties from the sites. Denote $\phi_p(x) = -\log(p(x)) = \sum_{i=1}^n \phi_i(x)$, then ϕ_p is $n\beta_m$ strongly-log-concave with its higher derivatives bounded.

Infinite Data limit

In an infinite data regime, Dehaene and Barthelmé [2015] have shown that an EP algorithm will behave like iterations of Newton's algorithm for finding the mode of a function. This behavior is then used to prove that EP is asymptotically exact. We will not go through the full derivation here, but shed some light into understanding the intuition behind the proof.

We will start by rephrasing Newton's update as an iterative method for finding Gaussian approximation to an arbitrary target distribution $p^*(x) \propto \exp\{-\psi(x)\}$. Recall in the classical setting, once we initialised at a point μ_1, Newton's method constructs a sequence of points with,

$$\mu_{t+1} = \mu_t - \psi^{''}(\mu_t)^{-1}\psi^{'}(\mu_t) \tag{9}$$

This is actually equivalent to the following Gaussian approximations of the target distribution. Starting from a Gaussian $g_1(x)$ with mean μ_1, we do the following

- Compute $\delta r_{t+1} = -\psi^{'(\mu_t)}$ and $\beta_{t+1} = \psi^{''}(\mu_t)$
- Compute a Gaussian approximation to $p^*(x)$:

$$g_{t+1} \propto \exp\left\{\delta r_{t+1}(x-\mu_t) - \beta_{t+1}\frac{(x-\mu_t)^2}{2}\right\} \tag{10}$$

$$\propto \exp\left\{(\delta r_{t+1} + \beta_{t+1}\mu_t)x - \frac{\beta_{t+1}}{2}x^2\right\} \tag{11}$$

- Compute the mean of $g_{t+1} : \mu_{t+1} = \mu_t - \psi^{''-1}\psi^{'}(\mu_t)$

We will now state the main result of Dehaene and Barthelmé [2015], which demonstrated that the EP updates converge towards the updates of the above Newton's algorithm,

Theorem 1 (limit behavior of EP) *Consider the EP approximation $(r_i, \beta_i)_{i\in[1,n]}$, in the limit of a large number of sites $n \to \infty with \mu_0$ constant where:*

1. *All cavity precisions are large, in the sense that $\min_i(\beta - \beta_i) = pn + \mathcal{O}(1)$ where p some positive constant and β the cavity mean*
2. *All hybrid h_i have similar means*

then, the asymptotic behaviour of the natural parameters of the global approximations obtained by the EP update is identical to the results of the Newton update starting from μ_0, and we have:

$$\sum_i r_i^{new} = -\psi^{'}(\mu_0) + (\sum_i \beta_i^{new})\mu_0 + \mathcal{O}(1) \tag{12}$$

$$\sum_i \beta_i^{new} = \psi^{''}(\mu_0) + \mathcal{O}(1) \tag{13}$$

Theorem 1 works when we are at a high precision regime in which the βs are large. In Dehaene and Barthelmé [2015], the authors investigated conditions for the high precision regime to be reached. In fact, they showed that around modes of the target distribution which are sufficiently peaked, the high precision regime is stable. Furthermore, this regime is attained in the classical large data limit, thus EP is asymptotically exact in the infinite data limit.

Finite Data Bounds

To illustrate the closeness of the first two moments of the target distribution and the EP approximation, we will utilise the result from Brascamp and Lieb [1976], which provides bounds on even moments of log-concave distributions. Dehaene and Barthelmé [2018] provided an extension to the above work, in that they bound the odd moments of log-concave distributions with slowly changing log-functions (as quantified by Eq (8)) and derive a first order Taylor expansion for the even moments. Here we will state the results of their extension, interested readers should refer to Dehaene and Barthelmé [2018] for detailed derivation. Denoting μ the true mean of the target distribution $p(x)$, we have the following bound:

$$|\phi_p^{'}(\mu)| \leq \frac{K_3}{2\beta_m} \tag{14}$$

which tells us that $\phi_p^{'}(\mu)$ is close to 0, thus μ should be close to the mode x^* of the distribution. This can be shown by setting:

$$\begin{aligned} |\phi_p^{'}(\mu)| &= |\phi_p^{'}(\mu) - \phi_p^{'}(x^*)| & (15)\\ &= |\phi_p^{''}(\xi)(\mu - x^*)| \qquad \xi \in [\mu, x^*] & (16)\\ &\geq |\phi_p^{''}(\xi)||\mu - x^*| & (17)\\ &\geq n\beta_m|\mu - x^*| \qquad \text{by } \beta_m \text{ strongly log concave assumption} & (18)\end{aligned}$$

Combining this with Eq(14), we obtain the following bound,

$$|\mu - x^*| \leq \frac{K_3}{2n\beta_m^2} \tag{19}$$

This quantifies how close the true mean and the mode of the target is. We now proceed to show that μ_{EP} is also close to x^* by applying the above reasoning to all hybrids $h_i(x)$:

$$\forall i \quad |\phi_i^{'}(\mu_{EP}) + \beta_{-i}\mu_{EP} - r_i| \leq \frac{K_3}{2n\beta_m} \tag{20}$$

since $\log(h_i(x)) = -\phi_i(x) - \beta_{-i}\frac{x^2}{2} + r_{-i}x$. Recall that $q(x|r, \beta)$ has mean μ_{EP}, thus we have $r = \beta\mu_{EP}$, which gives:

$$\begin{aligned} (\sum_i \beta_{-i})\mu_{EP} &= ((n-1)\beta)\mu_{EP} & (21)\\ &= (n-1)r & (22)\\ &= \sum_i r_{-i} & (23)\end{aligned}$$

Thus if we sum up every hybrid $h_i(x)$, the β_i and r_i cancel out each other. Now apply the triangle inequality to obtain the following,

$$|\phi_p^{'}(\mu_{EP})| \leq \frac{K_3}{2\beta_m} \tag{24}$$

thus we have μ_{EP}, similar to μ, is close to the mode x^*,

$$|\mu_{EP} - x^*| \leq \frac{K_3}{2n\beta_m^2} \tag{25}$$

Combining Eq(19) and Eq(25), we can show $\mu = \mu_{EP} + O(\frac{1}{n})$. Similarly, the quality of the approximation on the precision (inverse variance) can be shown to be related as $\sigma^{-2} = \sigma_{EP}^{-2} + O(1)$. In fact, Dehaene and Barthelmé [2018] further shown in his work that the bound can be even tighter,

$$|\mu - \mu_{EP}| \leq O(n^{-2}) \tag{26}$$

$$|\sigma^{-2} - \sigma_{EP}^{-2}| \leq O(n^{-2}) \tag{27}$$

Though these bounds are useful, they are coarse and fail to show significant improvement of EP over other approximation methods Dehaene and Barthelmé [2018]. Furthermore, they rely on strong assumptions that distributions are log-concave with slowly-changing log-functions, which is often unrealistic in practice. Further study of the asymptotic properties of EP is much needed.

3 Examples

3.1 Toy Example

We implemented the EP algorithm given as pseudocode by Rasmussen and Williams [2006], code is available on Github. We tested it on a simple two-dimensional classification problem. The data, shown in Figure 2 as green and red dots correspond to classes -1 and +1, respectively.

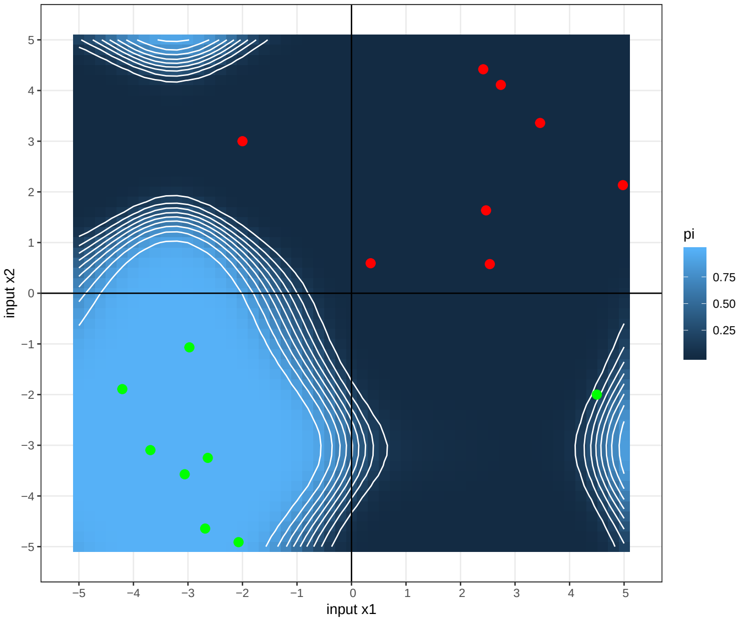

Figure 2: Two-dimensional toy classification dataset. Data points from class -1 (+1) are shown in green (red). Contour plot shows posterior predictive probability for class -1.

Figure 2 demonstrates that the points are classified appropriately. The posterior probability for class -1 is highest around the green points. It should be noted that the results appear to be very sensitive to the choice of hyperparameters for the covariance function. The choice above was made by maximising the marginal log likelihood computed by EP. resulting in $\sigma_f = 2.14, l = 1.54$.

3.2 Handwritten Digit Classification

We evaluate EP as an approximation method for binary classification, where the likelihood is non-Gaussian. Specifically, we look to replicate results from Chapter 3 of Rasmussen and Williams [2006], in which the authors classify handwritten digits from the US Postal Service database. The database contains 9,298 greyscale images of digits, segmented into 16x16 pixels and normalized so that the pixel intensity lies between [-1,1]. The database also contains 0-9 class labels for all digits. Figure 6 in the Appendix shows an example of the data used. The USPS data accessed from is divided into equally-sized training and test sets, divisions which we preserve here.

In this example, we focus on binary classification - distinguishing 2's from 9's - using the intensity values from the 256 pixels as predictors in an approximate Gaussian Process regression with a logistic likelihood. The likelihood in non-Gaussian, so is not conjugate with the GP prior, so the posterior cannot be calculated analytically. Instead, we use EP to approximate the posterior and compare the results to a Laplace approximation, an approximation with Variational Bayes (VB), an exact solution using MCMC and a simple logistic regression classifier with LASSO for variable selection and dimensionality reduction. The logistic regression was fit with the `glmnet` package in `R` using 5-fold cross-validation and the final model selected was that which had the minimum value of λ. All the other models were fit using the `GPML` Toolbox developed for `Matlab` and `Octave` by Rasmussen and Nickisch [2010].

3.2.1 Hyperparameters

In Gaussian Process regression, the outcome y is modeled as:

$$\begin{aligned} y &= f(\boldsymbol{x}) + \epsilon \\ \epsilon &\sim \mathcal{N}(0, \sigma_n^2) \\ f(\boldsymbol{x}) &\sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \end{aligned}$$

where $m(\boldsymbol{x})$ is the mean function of the process and $k(\boldsymbol{x}, \boldsymbol{x}')$ is the covariance function, or covariance kernel. For all models considered here, we use the zero mean function ($m(\boldsymbol{x}) = 0$) and the squared exponential covariance function for k with no noise variance (i.e. $\sigma_n^2 = 0$):

$$k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^2}{2\ell^2}\right)$$

The ℓ and σ_f^2 in the equation above are hyperparameters of the GP; ℓ is the length-scale parameter and σ_f^2 is the the signal variance. Fitting a GP regression involves tuning the hyperparameters using training data. Here we select hyperparameters for each approximation method to be the values that minimize the negative log marginal likelihood of the data.

Following a procedure similar to that used in Rasmussen and Williams [2006], we calculated the negative log likelihood for 15 evenly-spaced values of each hyperparameter, $\log(\sigma_f^2)$ and $\log(\ell)$ using the 897 training observations of 9's and 2's . Figure 3 shows the resulting contour plots. The points depicted in Figure 3 are the hyperparameter values resulting from a gradient descent minimization of the negative log marginal likelihood, a procedure defined in Rasmussen and Nickisch [2010].

The two optimization procedures resulted in similar optimal values of hyperparameters, and relatively similar values across approximation methods. The contour plot for EP hyperparameters is missing contours for a section of the parameter space - where the evaluation of the marginal likelihood failed. Many steps were taken in implementation to avoid numerical instability which is a known drawback of EP, and the ultimate reason for failure is not clear. Optimizing the hyperparameters for MCMC was computationally infeasible for this project, so the average of the optimal values of σ_f^2 and ℓ for other methods were used for the MCMC classifier. Optimizing hyperparameters for EP took 50 times as long as Laplace and 13 times as long as VB (exact times shown in Table 1).

3.2.2 Results

All classification methods performed quite well distinguishing “2”s from “9”s in the test set. Classification rates for the test set based on a hard cutoff of 1/2 for the predicted class probability are given in Table 1, and show that the worst-performing method, Logistic regression, still correctly classified over 99% of test cases. The Laplace approximation actually exhibited the highest classification rate, higher even than EP or MCMC, though by a small margin.

Table 1 also gives the computation time needed to estimate the hyperparameters and fit each classifier. The computation time required for MCMC is several orders of magnitude larger than that required for any of the other methods, and in this case, did not produce enough of an improvement in classification performance to warrant the computation cost. In addition to having the highest test set classification rate, the GP with Laplace approximation was also by far the fastest GP method, though still slower than logistic regression

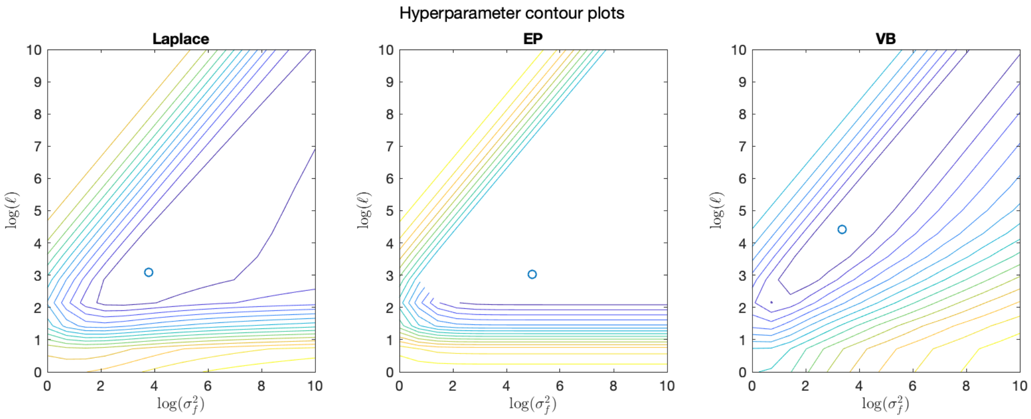

Figure 3: Contour plots of negative log marginal likelihood values as a function of GP hyperparameters $\log(\sigma_f^2)$ and $\log(\ell)$

Method	Classification rate	Hyperparams time (sec)	Prediction time (sec)	Total time (sec)
Logistic regression	99.06%	-	-	2.73
Laplace	99.41%	13.3	0.23	13.53
EP	99.30%	678.95	6.56	685.51
VB	99.08%	51.56	1.72	53.28
MCMC	99.18%	-	308.32	$>$ 308.32

Table 1: Classification rates (in the test set) and computation time for each classification method. Hyperparameters were not optimized for logistic regression or MCMC due to no hyperparameters needed and large computation time, respectively.

with LASSO variable selection. This particular application appears simple enough that simple methods with strong assumptions like logistic regression and a GP with Laplace approximation perform quite well.

Figure 4 shows the distributions of the latent means (before the latent function value has been transformed by the logistic function) and the predicted means. Though all the methods have similar bimodal distributions, the distribution of EP latent means spans (-100,200), a scale of an order of magnitude greater than the other classifiers' latent means. As a result, EP has more extreme, or "harder" (close to -1 and 1) predicted means than the other methods, while Laplace has the most moderate predicted means. This finding is similar to that of Rasmussen and Williams [2006].

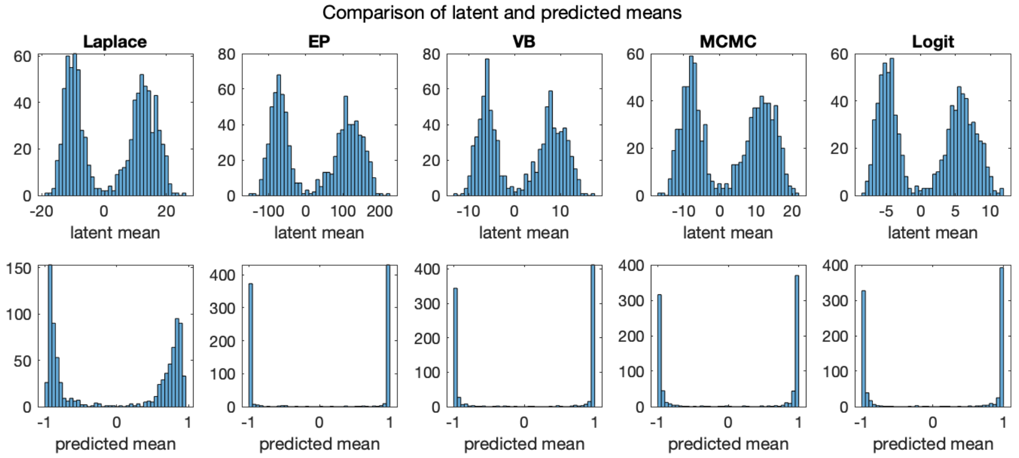

Figure 4: Latent (top row) and predicted (bottom row) means for each classifier. Higher values correspond to higher predicted probability that the observation is a "2" instead of a "9".

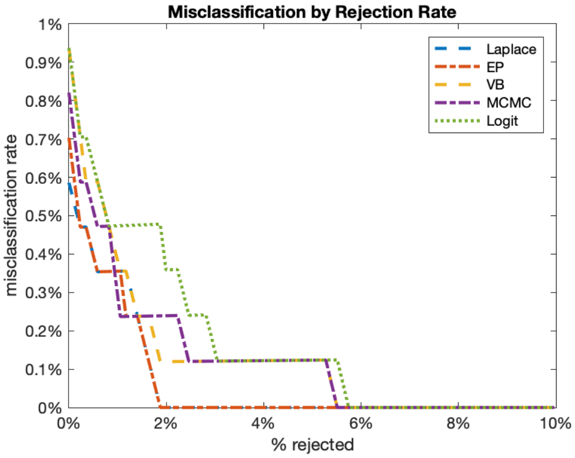

Figure 5: Misclassification rate plotted as a function of % of predictions rejected

Thus far, we have classified observations by selecting the class with the highest probability. In order to further highlight differences between the methods, we also examine classification rates based of a requirement that class probabilities exceed a certain threshold. This results in a subset of observations being "rejected," or remaining unclassified. In a practical setting, those observations would be revisited and classified with a more complex method. Figure 5 shows the missclassification rate as a function of the proportion of observations rejected for each classifier. EP and Laplace perform almost identically, and outperform the other methods. Logistic regression seems to perform the worst based on this criteria, being the slowest to reach perfect classification.

4 Discussion

Expectation propagation is a flexible method for approximating complex and intractable integrals. It offers better computational speed than MCMC, while providing better accuracy than other approximation methods such as Laplace or Variational Bayes.

EP has a number of advantages. EP can be generalised relatively easily past Gaussian priors, to other exponential family distributions Barthelmé [2016]. EP appears to scale relatively well in terms of computational complexity. The most computationally intensive aspect of EP is the numerical integration needed to compute moments of the marginals. However this is relatively easy Barthelmé [2016]. For binary Gaussian process classification specifically, there are results that show EP offers much better results than techniques such as Variational Bayes or Laplace approximation [Nickisch and Rasmussen, 2008].

However, it is difficult to obtain theoretical results for EP, for instance proofs of convergence. EP is slower than Variational Bayes or Laplace approximation techniques, due to the numerical integration involved. Implementation of EP can be difficult, particularly for complex problems. It is a fixed point algorithm, but numerical instabilities can arise causing it to explode. For instance, covariance matrices can accumulate noise over the iterations and result in eigenvalues going to 0 [Barthelmé, 2016]. Careful implementation of methods such as Cholesky factorisation, or likelihood tempering [Minka, 2004] is needed.

We have demonstrated that EP performed well for the binary digit classification task; although compared to simpler methods, a very small gain in accuracy was significantly outweighed by the longer runtime. Optimisation of hyperparameters using training data was particularly computationally intensive.

References

Simon Barthelmé. The Expectation-Propagation algorithm: a tutorial – Part 1. CIRM Audiovisual resource, 2016.

Herm Jan Brascamp and Elliott H Lieb. Best constants in young's inequality, its converse, and its generalization to more than three functions. *Advances in Mathematics*, 20(2):151–173, 1976.

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

Guillaume P Dehaene and Simon Barthelmé. Bounding errors of expectation-propagation. In *Advances in Neural Information Processing Systems*, pages 244–252, 2015.

Thomas Minka. Power ep. Technical report, Technical report, Microsoft Research, Cambridge, 2004.

Thomas 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.

Hannes Nickisch and Carl Edward Rasmussen. Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015, 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes in Machine Learning*. the MIT Press, 2006. ISBN 0-262-18253-X. URL `http://www.gaussianprocess.org/gpml/chapters/RW.pdf`.

A Digit Classification Appendix

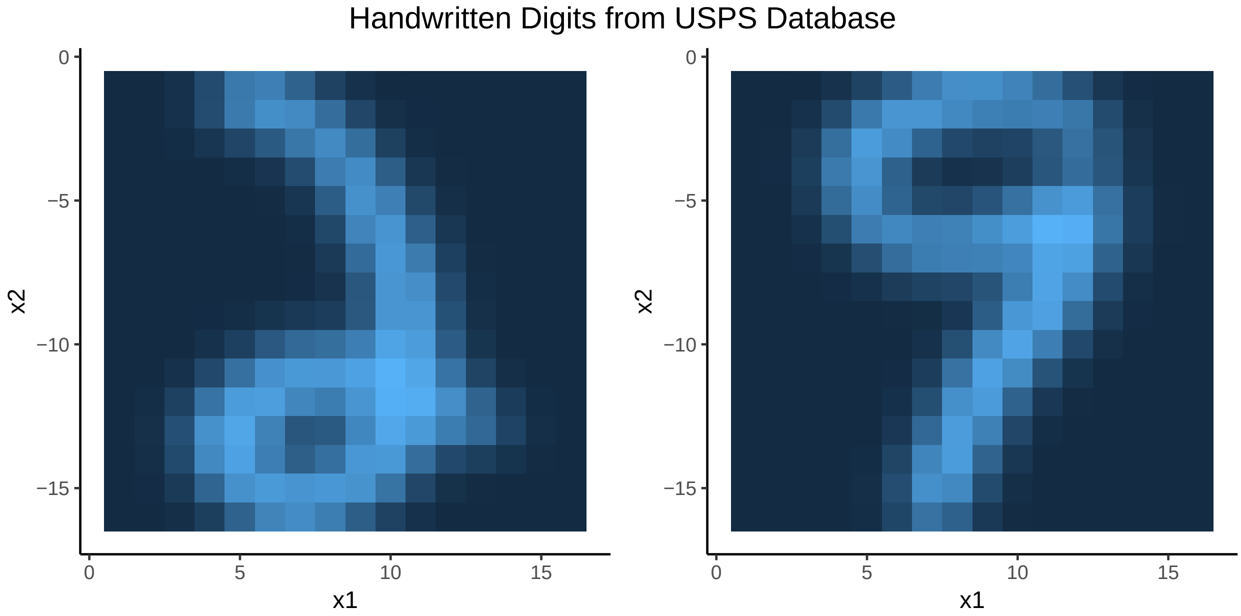

Figure 6: Example of a handwritten ”2” (left) and ”9” (right) from the USPS database. Images are segmented into 16x16 pixels and scaled to have intensity [-1,1].

x1	x2	Intensity of “2”	Intensity of “9”
-1	1	-1.00	-1.00
-1	10	-0.99	0.18
-1	11	-1.00	-0.09
-1	12	-1.00	-0.48
-1	13	-1.00	-0.84
-1	14	-1.00	-0.98

Table 2: Example of intensity data for selected handwritten digits from the USPS dataset corresponding to the digits depicted in Figure 6.

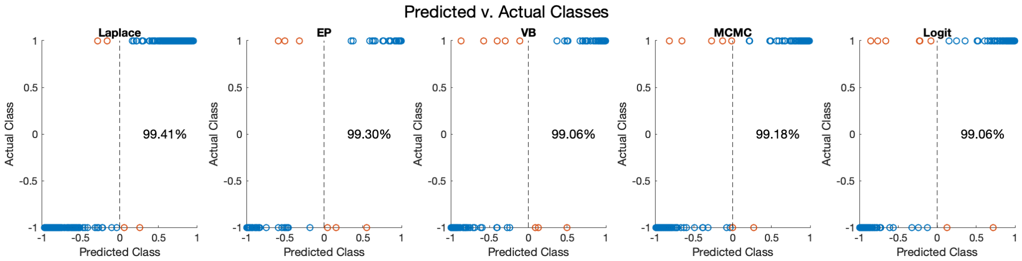

Figure 7: Relationship between predicted and actual classes in the test set for each classifier. Correct classifications are shown in blue, while misclassifications are red. Overall classification rates are shown on the plot.