Robust Expectation Propagation in Factor Graphs Involving Both Continuous and Binary Variables

Marco Cox
Department of Electrical Engineering
Eindhoven University of Technology
m.g.h.cox@tue.nl

Bert de Vries
Department of Electrical Engineering
Eindhoven University of Technology
and GN Hearing, Eindhoven, The Netherlands
bdevries@ieee.org

***Abstract*—Factor graphs provide a convenient framework for automatically generating (approximate) Bayesian inference algorithms based on message passing. Examples include the sum-product algorithm (belief propagation), expectation maximization (EM), expectation propagation (EP) and variational message passing (VMP). While these message passing algorithms can be generated automatically, they depend on a library of pre-computed message update rules. As a result, the applicability of the factor graph approach depends on the availability of such rules for all involved nodes. This paper describes the probit factor node for linking continuous and binary random variables in a factor graph. We derive (approximate) sum-product message update rules for this node through constrained moment matching, which leads to a robust version of the EP algorithm in which all messages are guaranteed to be proper. This enables automatic Bayesian inference in probabilistic models that involve both continuous and discrete latent variables, without the need for model-specific derivations. The usefulness of the node as a factor graph building block is demonstrated by applying it to perform Bayesian inference in a linear classification model with corrupted class labels.**

I. Introduction

Probabilistic programming has gained a lot of interest in recent years, leading to the development of a range of software packages such as Stan [1], Edward [2], ZhuSuan [3] and others. The idea behind probabilistic programming is to completely automate the derivation of an inference algorithm: the user only has to specify the probabilistic model, and the software will automatically generate an inference algorithm to calculate the (approximate) posterior distributions of interest. Historically, such systems have relied on Markov chain Monte Carlo (MCMC) techniques, which are rather slow and as a result do not scale well to large models. In recent years, black-box variational inference (BBVI) algorithms [4], [5] have been developed that are much more efficient, leading to a renewed interest in the field.

Another line of work towards automating inference in probabilistic models has focussed on graphical models, and in particular on factor graphs. A factor graph is a graphical representation of any probabilistic model, and it is useful from an algorithmic point of view because a lot of inference algorithms can be formulated as message passing on such graphs [6]. Since these message passing algorithms can be derived automatically, factor graphs naturally lend themselves to automatic derivation of inference algorithms. Examples of such algorithms include the sum-product algorithm (belief propagation) for exact Bayesian inference and expectation maximization (EM) [7], variational message passing (VMP) [8], expectation propagation (EP) [9], and particle filtering for approximate inference. Factor graphs are particularly well suited for describing time series models such as state-space models of dynamical systems and hidden Markov models. In such models, message passing algorithms may recover well known algorithms such as the Kalman filter/smoother, the Viterbi algorithm or the Baum-Welch algorithm without the need to manually derive them. Loeliger [10] provides an excellent introduction of factor graphs in the context of signal processing.

Message passing algorithms, MCMC methods and BBVI algorithms all share the same goal: providing an automatable way to derive an inference algorithm. Which method is most appropriate depends on the model at hand. Message passing algorithms depend on a library of so-called message update rules: reusable but manually derived update equations for the required messages. BBVI on the other hand does not require any manual derivations, but suffers from other drawbacks. Just like MCMC, it is generally significantly slower than analytical message passing algorithms since the black-box approach does not allow the exploitation of known analytical solutions to the involved integrals. This is especially a problem in the signal processing setting, where we generally have models with lots of variables due to time unrolling. Where message passing algorithms are often fast enough to be executed in real-time, this is usually not the case with BBVI. Moreover, BBVI in its standard form cannot handle models that involve discrete latent variables.

Our goal in this paper is to extend the applicability of efficient message passing algorithms to models involving both continuous and binary (latent) variables, linked through the probit link function. Towards this end, we derive approximate sum-product update rules for the probit link factor node based on moment matching, which leads to the well known EP message passing algorithm [9], [11]. In case the binary random variable is observed, the obtained EP algorithm is a familiar one; it has been derived for example for Bayesian linear probit regression, Gaussian process classification [12] and for inference in the TrueSkill model for rating gamers [13]. However, in this work we consider the more general

ISBN 978-90-827970-1-5

case in which the binary variable is *latent*. We show that this can lead to improper messages, and we describe a principled way to modify the approximation such that all messages are guaranteed to be proper, yielding a robust version of the EP algorithm.

The introduced factor node and its corresponding message update rules can be used as an "off-the-shelf" building block, removing the need for model-specific derivations. We demonstrate its use by performing Bayesian learning of a linear classifier from corrupted class labels through message passing on the corresponding factor graph.

II. Related work

Ziniel et al. [14] describe binary linear classification in a factor graph by generalized approximate message passing, an approach that closely resembles expectation propagation. Both the logistic and probit link functions are considered, but the derived messages are not guaranteed to be proper in case the binary variable is latent. The TrueSkill™ rating system for multi-player games described in [13] uses a factor graph model with a node similar to the one described in this work to link the continuous skill levels of two players to the binary outcome of a match. However, the derivation assumes the binary variable to be observed. Hu et al. [15] consider a different factor node to link continuous and binary variables. They propose to solve the problem of improper messages taking an empirically tuned mixture with a proper message.

III. Forney-style factor graphs

To set the stage for the rest of the paper, we briefly introduce (Forney-style) factor graphs. A factor graph is an undirected graph that encodes a factorization of a function, for example of a joint probability density function (PDF). Since a generative probabilistic model is nothing more than a joint PDF over all model variables, any generative probabilistic model can be represented as a factor graph. A Forney-style factor graph (FFG) is a specific type of factor graph in which nodes represent factors and edges represents variables. Consider the following simple example:

$$p(a,b,c,d) = p(a,b)\, p(b,c,d)\, p(b).$$

This joint PDF consists of three factors, so one might expect the corresponding FFG to consist of 3 nodes. However, since b is involved in more than two factors, it cannot be represented by a single edge. This issue is resolved by introducing auxiliary variables b' and b'' as well as an "equality constraint factor" $p(b, b', b'') = \delta(b' - b)\,\delta(b'' - b)$. The equivalent augmented joint PDF becomes:

$$p(a,b,c,d,b',b'') = p(a,b)\, p(b',c,d)\, p(b'')\, p(b,b',b''),$$

in which every variable appears in at most two factors. Figure 1 depicts the resulting FFG. It is always possible to find such an augmented factorization, so any factorized PDF can be represented as an FFG.

A factor node implements a known (stochastic or deterministic) relationship among the variables connected to it, such as addition: $p(a,b,c) = \delta(a+b-c)$. Probabilistic inference of latent variables in a factor graph is achieved by "clamping" observed variables to their observed values, and then propagating sum-product (belief) messages along the edges of the graph [16]. These messages are calculated from only local information: incoming messages at the same node and the node function. We refer to [17] for a more elaborate introduction to FFGs and (sum-product) message passing for probabilistic inference.

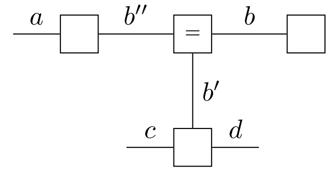


Figure 1: Forney-style factor graph representation of joint PDF $p(a,b,c,d) = p(a,b)\,p(b,c,d)\,p(b)$. Nodes represent factors and edges correspond to variables.

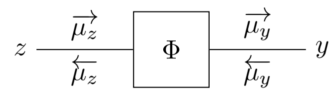


Figure 2: Forney-style factor graph (FFG) representation of the probit factor node. The symbols over and below the edges represent inbound and outbound messages, which are generally (unnormalized) probability distributions.

IV. The probit link factor node

The probit factor node links a continuous random variable $z \in \mathbb{R}$ to binary random variable $y \in \{-1, +1\}$ through the cumulative density function of the standard Gaussian distribution: $P(y|z) = \Phi(y \cdot z)$, with $\Phi(x) \triangleq \int_{-\infty}^{x} \mathcal{N}(t|0,1)\mathrm{d}t$. We are interested in the (approximate) sum-product messages corresponding to this factor, as depicted in Figure 2.

A. From real to binary

Let $\overrightarrow{\mu_z}$ be an incoming Gaussian sum-product message:

$$\overrightarrow{\mu_z}(z) = \mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z}). \tag{1}$$

The outbound sum-product message towards y is obtained by applying the sum-product rule [16]:

$$\begin{aligned} \overrightarrow{\mu_y}(y) &= \int_{-\infty}^{\infty} \overrightarrow{\mu_z}(z)\Phi(z \cdot y)\mathrm{d}z \\ &= \int_{-\infty}^{\infty} \mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z})\Phi(z \cdot y)\mathrm{d}z \\ &= \Phi\left(\frac{y \cdot \overrightarrow{m_z}}{\sqrt{1+\overrightarrow{v_z}}}\right), \end{aligned} \tag{2}$$

where we used the solution to the Gaussian integral as derived in [12]. Note that the symmetry of Φ ensures that $\overrightarrow{\mu_y}$ is a valid probability mass function on $\{-1, +1\}$.

ISBN 978-90-827970-1-5 © EURASIP 2018

B. From binary to real

In the other direction, incoming sum-product message $\overleftarrow{\mu_y}$ is a probability mass function with "odds" parameter α:

$$\overleftarrow{\mu_y}(y) = \begin{cases} (1-\alpha) & \text{if } y = -1, \\ \alpha & \text{if } y = +1. \end{cases} \tag{3}$$

If y is observed, α is either 0 or 1, but we consider the general case $\alpha \in [0, 1]$. The exact sum-product message towards z follows again from the sum-product rule:

$$\begin{aligned} \overleftarrow{\mu_z}(z) &= \sum_{y \in \{-1,+1\}} \overleftarrow{\mu_y}(y)\Phi(z \cdot y) \\ &= (1-\alpha)\Phi(-z) + \alpha\Phi(z) \\ &= (1-\alpha)(1-\Phi(z)) + \alpha\Phi(z) \\ &= 1 - \alpha + (2\alpha - 1)\Phi(z). \end{aligned} \tag{4}$$

This message is not a valid (unnormalized) probability distribution, which is problematic if it is to be passed along through the factor graph. Moreover, since the marginal distribution of z is per definition proportional to the product of sum-product messages $\overrightarrow{\mu_z}$ and $\overleftarrow{\mu_z}$, it implies a non-Gaussian marginal:

$$\begin{aligned} p_z(z) &\propto \overrightarrow{\mu_z}(z) \cdot \overleftarrow{\mu_z}(z) \\ &= (1-\alpha)\mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z}) + (2\alpha-1)\mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z})\Phi(z). \end{aligned} \tag{5}$$

If $\overleftarrow{\mu_y}$ is uninformative ($\alpha = 0.5$), p_z reduces to the incoming message $\overrightarrow{\mu_z}$, which is to be expected. If $\overleftarrow{\mu_y}$ has zero variance ($\alpha = 0$ or $\alpha = 1$), p_z is an asymmetric "tilted" distribution. In other cases p_z might be bimodal.

V. Robust expectation propagation

To obtain a feasible message passing algorithm, we can approximate the 'problematic' exact sum-product message $\overleftarrow{\mu_z}$ by a Gaussian message $\overleftarrow{\nu_z} \approx \overleftarrow{\mu_z}$. There are multiple ways to calculate the approximate message. In this work we will calculate $\overleftarrow{\nu_z}$ by first approximating the exact marginal distribution p_z by a Gaussian distribution q_z that minimizes the Kullback-Leibler divergence $D_{\mathrm{KL}}(p_z||q_z)$. Then, $\overleftarrow{\nu_z}$ is calculated through the implied marginal property from Eqn. 5: $q_z \propto \overrightarrow{\mu_z} \cdot \overleftarrow{\nu_z} \Longrightarrow \overleftarrow{\nu_z} \propto q_z / \overrightarrow{\mu_z}$. Since q_z and $\overrightarrow{\mu_z}$ are both Gaussians, $\overleftarrow{\nu_z}$ is also Gaussian and its parameters can be expressed analytically.

Replacing the exact sum-product message $\overleftarrow{\mu_z}$ with an approximate message $\overleftarrow{\nu_z}$ and passing the approximate message along through the factor graph as if it were a regular sum-product message yields a message passing implementation of the EP algorithm [9], [11]. Since the approximate message depends on incoming message $\overrightarrow{\mu_z}$ – referred to as the "cavity distribution" that carries information about z from other parts of the graph – the resulting message passing schedule contains circular dependencies and will require multiple passes to converge.

A. Derivation of the EP message

Because q_z is chosen to be in the exponential family, $D_{\mathrm{KL}}(p_z||q_z)$ is minimized when the moments of q_z are equal to those of p_z. For the probit link factor, these moments can be expressed analytically. From Eqn. 5 we have:

$$p_z(z) = C^{-1}[(1-\alpha)\mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z}) + (2\alpha-1)\underbrace{\mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z})\Phi(z)}_{g(z)}].$$

The first and second order moments of g can be derived in a similar fashion as in Section 3.9 of [12], and we just state the results here:

$$\begin{aligned} \mu_g^{(1)} &= \Phi(\gamma)\overrightarrow{m_z} + \frac{\overrightarrow{v_z}\mathcal{N}(\gamma|0,1)}{\sqrt{1+\overrightarrow{v_z}}}, \\ \mu_g^{(2)} &= 2\overrightarrow{m_z}\mu_g^{(1)} + (\overrightarrow{v_z} - \overrightarrow{m_z}^2)\Phi(\gamma) + \frac{\overrightarrow{v_z}^2\gamma\mathcal{N}(\gamma|0,1)}{1+\overrightarrow{v_z}}, \end{aligned}$$

with $\gamma = \frac{\overrightarrow{m_z}}{\sqrt{1+\overrightarrow{v_z}}}$. Normalization constant C is obtained by integration:

$$C = 1 - \alpha + (2\alpha - 1)\int_{-\infty}^{\infty} g(z)\mathrm{d}z = 1 - \alpha + (2\alpha - 1)\Phi(\gamma).$$

Finally, the moments of p_z are easily expressed in terms of the moments of g:

$$\begin{aligned} \mu_{p_z}^{(1)} &\triangleq \int_{-\infty}^{\infty} z \cdot p_z(z)\mathrm{d}z = C^{-1}\left[(1-\alpha)\overrightarrow{m_z} + (2\alpha-1)\mu_g^{(1)}\right], \\ \mu_{p_z}^{(2)} &\triangleq \int_{-\infty}^{\infty} z^2 \cdot p_z(z)\mathrm{d}z \\ &= C^{-1}\left[(1-\alpha)(\overrightarrow{m_z}^2 + \overrightarrow{v_z}) + (2\alpha-1)\mu_g^{(2)}\right]. \end{aligned}$$

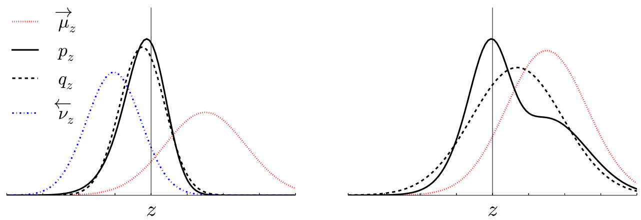
The Gaussian approximation to the exact marginal p_z becomes $q_z(z) = \mathcal{N}(z|\tilde{m}_z, \tilde{v}_z)$ where $\tilde{m}_z = \mu_{p_z}^{(1)}$ and $\tilde{v}_z = \mu_{p_z}^{(2)} - (\mu_{p_z}^{(1)})^2$. Once q_z is known, the approximate message $\overleftarrow{\nu_z}$ follows by dividing q_z by $\overrightarrow{\mu_z}$:

$$\begin{aligned} &\underbrace{\mathcal{N}(z|\tilde{m}_z, \tilde{v}_z)}_{q_z(z)} \propto \underbrace{\mathcal{N}(z|\overrightarrow{m_z}, \overrightarrow{v_z})}_{\overrightarrow{\mu_z}(z)} \cdot \underbrace{\mathcal{N}(z|\overleftarrow{m_z}, \overleftarrow{v_z})}_{\overleftarrow{\nu_z}(z)} \\ &\Rightarrow \begin{cases} \overleftarrow{v_z} &= (\tilde{v}_z^{-1} - \overrightarrow{v_z}^{-1})^{-1}, \\ \overleftarrow{m_z} &= \overleftarrow{v_z}(\tilde{v}_z^{-1}\tilde{m}_z - \overrightarrow{v_z}^{-1}\overrightarrow{m_z}). \end{cases} \end{aligned} \tag{6}$$

Note that the approximate message depends only on node-local information: the node function and both incoming sum-product messages. Figure 3 illustrates the approximation for two values of α. If $\alpha \in \{0, 1\}$, the variance of p_z is upper bounded by the variance of $\overrightarrow{\mu_z}$, so the approximate message $\overleftarrow{\nu_z}$ is guaranteed to be proper. In other cases, p_z can become bimodal, and its variance might exceed that of $\overrightarrow{\mu_z}$. If this happens, the approximate message would be an improper Gaussian with 'negative variance'.

B. Handling improper messages

If the variance of marginal distribution p_z is larger than the variance of incoming cavity message $\overrightarrow{\mu_z}$, approximate message $\overleftarrow{\nu_z}$ is not a proper probability distribution ($\overleftarrow{v_z} < 0 \Rightarrow \int_{-\infty}^{\infty} \overleftarrow{\nu_z}(z)\mathrm{d}z > 1$). This can be a problem if the



(a) $\alpha = 0$ (b) $\alpha = 0.1$

Figure 3: Illustration of the moment matching approximation $q_z \approx p_z$ and the resulting approximate sum-product message $\overleftarrow{\nu_z}$. (a): $\overleftarrow{\mu_y}(y) = \delta(y+1)$ (y is observed), resulting in a proper approximate message. (b): $\overleftarrow{\mu_y}(y) = \text{Bernoulli}(y|0.1)$ (y is a latent variable). The variance of q_z is larger than the variance of $\overrightarrow{\mu_z}$, leading to an improper approximate message with 'negative variance'.

message is to be passed along through the factor graph, since it can lead to improper cavity and marginal distributions. Previously proposed ways to handle such messages include: (a) setting the variance of $\overleftarrow{\nu_z}$ to infinity, (b) setting the variance of $\overleftarrow{\nu_z}$ to its absolute value [18], and (c) replacing the improper message with an empirically tuned mixture of itself and another message [15].

We propose a different, principled method to ensure properness. If properness of all messages is required, we replace the moment matching Gaussian approximation q_z with a Gaussian approximation that minimizes $D_{\text{KL}}(p_z||q_z)$ under the constraint $\text{var}[q_z] \leq \text{var}[\overrightarrow{\mu_z}] - \epsilon$ for some small ϵ. This leads to $\tilde{v}_z = \min\{\mu_{p_z}^{(2)} - (\mu_{p_z}^{(1)})^2, \overrightarrow{v_z} - \epsilon\}$. Under this approximation, the approximate message is guaranteed to be proper, leading to a robust EP algorithm. For numerical stability it is best to parametrize the approximate Gaussian message by its natural parameters:

$$\begin{aligned} \overleftarrow{\nu_z}(z) &\propto \exp(-\tfrac{1}{2}\overleftarrow{w_z}z^2 + \overleftarrow{\xi_z}z), \\ \overleftarrow{w_z} &= \tilde{v}_z^{-1} - \overrightarrow{v_z}^{-1}, \\ \overleftarrow{\xi_z} &= \tilde{v}_z^{-1}\tilde{m}_z - \overrightarrow{v_z}^{-1}\overrightarrow{m_z}. \end{aligned} \tag{7}$$

VI. Application: Linear Classifier with Corrupted Labels

To demonstrate an application of the described factor node, we use it to perform automatic Bayesian inference in a linear classifier with corrupted binary class labels. Assume the following generative model with feature vector $\mathbf{x}$, weight vector $\boldsymbol{\beta}$, and observed class label y:

$$\begin{aligned} \boldsymbol{\beta} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \\ z_i|\mathbf{x}_i, \boldsymbol{\beta} &\sim \text{Bernoulli}(\Phi(\mathbf{x}_i^T\boldsymbol{\beta})), \\ y_i|z_i, \alpha &\sim \text{Bernoulli}(\alpha^{\delta(z_i-1)} + (1-\alpha)^{\delta(z_i+1)}). \end{aligned} \tag{8}$$

Under this model, the 'true' class label z is a latent variable, and the observed label y is a corrupted version of z with symmetric label flipping probabilities α. If N data points

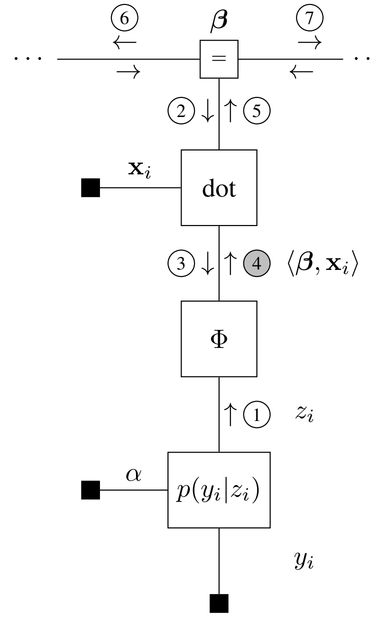
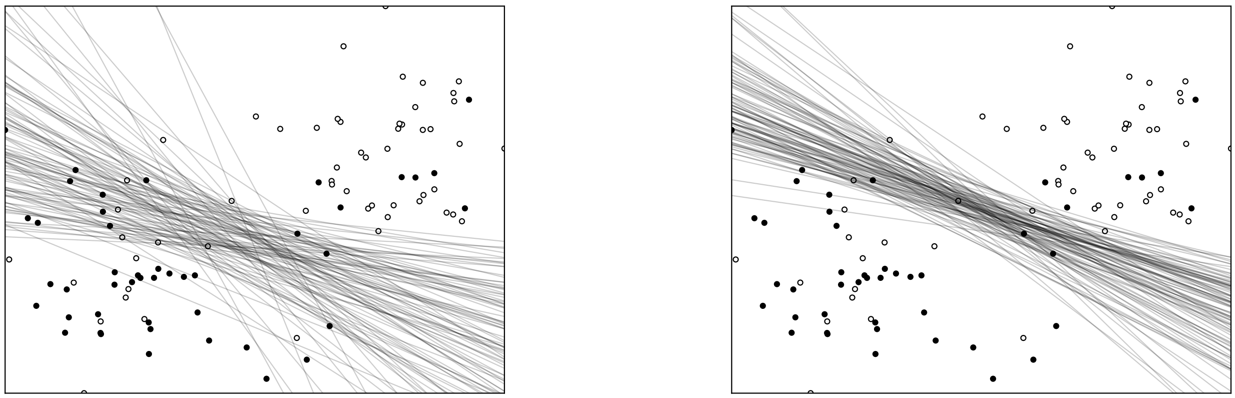


Figure 4: A section of the FFG representation of the linear binary classification model from Eqn. 8. Solid black boxes indicate clamping of a variable to an observed value. The circled numbers depict an EP message passing schedule. Approximate message ④ is calculated from Bernoulli message ① (information from the i-th data point) and Gaussian "cavity message" ③ (information from all other data points). ④ is then passed along through the factor graph like a regular sum-product (belief) message. The algorithm requires multiple iterations over all sections to converge since ④ depends on ③, which in turn depends on the approximate messages in other sections.

$\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ are observed, the FFG representation of this model consists of a node for $p(\boldsymbol{\beta})$ together with N identical graph sections as depicted in Figure 4. The posterior distribution of the weight vector, $p(\boldsymbol{\beta}|\mathcal{D})$, can be approximated by a multivariate Gaussian by executing an EP message passing schedule on the factor graph. The circled numbers in Figure 4 illustrate such a schedule, which can be generated automatically and only depends on a library of pre-computed message update rules. The Gaussian message update rules for the equality constraint node and dot product node are given for example in [19] (the dot product node is a special case of the matrix multiplication node). The update rule for message ① is trivial since y and z are both binary variables.

To test the depicted EP algorithm, we performed inference on a range of synthetic data sets with varying label corruption probabilities. Figure 5 shows results for a 2-dimensional data set generated with corruption probability 0.2. If α is set to 0, our model reduces to the standard probit regression model. To test the usefulness of modeling label corruption, we performed inference for both $\alpha = 0$ and $\alpha = 0.2$ (the true value). As is clear from the figures, the corruption-free model struggles to

ISBN 978-90-827970-1-5



(a) Result for $\alpha = 0$ (corruption-free model) (b) Result for $\alpha = 0.2$ (correct corruption model)

Figure 5: Inference results on a synthetic data set containing 100 points generated with label corruption probability 0.2. Lines are sampled from the inferred posterior distribution of the decision boundary.

explain the corrupted data points, leading to a lower model evidence. Setting the corruption parameter to the value used to generate the data consistently leads to better results.

VII. Conclusions

The described probit factor node can be used as a building block to link continuous and binary variables in a factor graph. We derived message update rules that enable fast EP-based inference in probabilistic models where both the continuous and binary variable are latent, without the need for model-specific derivations. While convergence of the EP algorithm is not guaranteed, our experiments show consistent and fast convergence in practice. Thanks to the modularity of message passing algorithms, it is easy to combine EP with other message passing algorithms such as VMP, EM, or particle filtering. In contrast to pure black box inference algorithms, such message passing algorithms can exploit local model structure to obtain faster algorithms, resorting only to sampling-based methods in parts where no analytical solutions are available.

References

[1] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A. Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software*, vol. 20, no. 2, pp. 1–37, 2016.

[2] Dustin Tran, Matt Hoffman, Rif Saurous, Eugene Brevdo, Kevin P Murphy, and David Blei, "Deep Probabilistic Programming," in *Proceedings of the International Conference on Learning Representations*, 2017.

[3] Jiaxin Shi, Jianfei Chen, Jun Zhu, Shengyang Sun, Yucen Luo, Yihong Gu, and Yuhao Zhou, "ZhuSuan: A Library for Bayesian Deep Learning," *arXiv:1709.05870 [cs, stat]*, Sept. 2017, arXiv: 1709.05870.

[4] Rajesh Ranganath, Sean Gerrish, and David Blei, "Black Box Variational Inference," in *PMLR*, Apr. 2014, pp. 814–822.

[5] Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David M. Blei, "Automatic Variational Inference in Stan," *arXiv:1506.03431 [stat]*, June 2015, arXiv: 1506.03431.

[6] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, Inc., 2006.

[7] J. Dauwels, S. Korl, and H.-A. Loeliger, "Expectation maximization as message passing," in *International Symposium on Information Theory, 2005. ISIT 2005. Proceedings*, Sept. 2005, pp. 583–586.

[8] J. Dauwels, "On Variational Message Passing on Factor Graphs," in *IEEE International Symposium on Information Theory, 2007. ISIT 2007*, June 2007, pp. 2546–2550.

[9] Thomas P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proceedings of the 17th conference on Uncertainty in artificial intelligence*. 2001, pp. 362–369, Morgan Kaufmann Publishers Inc.

[10] Hans-Andrea Loeliger, Justin Dauwels, Junli Hu, Sascha Korl, Li Ping, and Frank R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, June 2007.

[11] Andrew Gelman, Aki Vehtari, Pasi Jylänki, Tuomas Sivula, Dustin Tran, Swupnil Sahai, Paul Blomstedt, John P. Cunningham, David Schiminovich, and Christian Robert, "Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data," *arXiv:1412.4869 [stat]*, Dec. 2014, arXiv: 1412.4869.

[12] Carl Edward Rasmussen and Christopher K. I Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[13] Ralf Herbrich, Tom Minka, and Thore Graepel, "Trueskill: A Bayesian skill rating system," in *Advances in neural information processing systems*, 2006, pp. 569–576.

[14] J. Ziniel, P. Schniter, and P. Sederberg, "Binary Linear Classification and Feature Selection via Generalized Approximate Message Passing," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 2020–2032, 2015.

[15] Junli Hu, Hans-Andrea Loeliger, Frank R. Kschischang, and Justin Dauwels, "A general computation rule for lossy summaries/messages with examples from equalization," in *Proceedings of the 44th Allerton Conference on Communication, Control, and Computing*, July 2006, pp. 27–29.

[16] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[17] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[18] Peng Sun, Chuanzong Zhang, Zhongyong Wang, Carles N. Manchón, and Bernard H. Fleury, "Iterative receiver design for ISI channels using combined belief-and expectation-propagation," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1733–1737, 2015.

[19] Sascha Korl, *A factor graph approach to signal modelling, system identification and filtering*, Ph.D. thesis, Swiss Federal Institute of Technology, Zurich, 2005.