Probability Learning and Soft Quantization in Bayesian Factor Graphs

Francesco A.N. Palmieri and Alberto Cavallo

Dipartimento di Ingegneria Industriale e dell'Informazione
Seconda Universitá di Napoli (SUN)
via Roma 29, 81031 Aversa (CE), Italy
{francesco.palmieri,alberto.cavallo}@unina2.it

Abstract. We focus on learning the probability matrix for discrete random variables in factor graphs. We review the problem and its variational approximation and, via entropic priors, we show that soft quantization can be included in a probabilistically-consistent fashion in a factor graph that learns the mutual relationship among the variables involved. The framework is explained with reference the "Tipper" example and the results of a Matlab simulation are included.

Keywords: Machine Learning, Factor Graphs, Bayesian Methods.

1 Introduction

Probability propagation on graphs is a very promising emerging paradigm for building intelligent signal processing systems [12]. Algorithms and applications are under development in many areas of research that range from communication and coding to signal processing and control. However, full use and development of artificial intelligence systems that operate with probability propagation techniques require refinements on a number of critical issues. Some of these are: 1. Propagation in graphs with cycles [1]; 2. Parameter learning [8]; 3. Graph-structure learning [13]; 4. Propagation and learning in hybrid graphs with both continuous and discrete variables; etc. In this paper we focus on learning the probability matrix in discrete-variable factor graphs [7][6] pointing to a connection to variational learning [5][3][2][18][19]. We apply the idea to a generic block where the whole probability matrix is learned from examples. Recent development on inference based on entropic priors [15][14] allows the introduction of soft quantization within the Bayesian graph framework much like in fuzzy logic [17]. Entropic priors allow to translate some of the successful heuristics typical of the fuzzy framework, into a probabilistically-consistent Bayesian learning paradigm on factor graphs. Soft logic formulated within standard probability theory [10] coupled with belief propagating on factor graphs represents a very promising framework to bring to a higher cognitive level many of the current signal processing problems. In our formulation we use factor graphs in Forney's normal

B. Apolloni et al. (Eds.): *Neural Nets and Surroundings*, SIST 19, pp. 3–10.
DOI: 10.1007/978-3-642-35467-0_1

form [11], because they are easier to handle in comparison to more traditional Bayesian graphs [16].

In this paper we first review the problem of learning the probability matrix pointing to a connection with variational message passing. Then we briefly introduce soft quantization with entropic priors and finally we apply the ideas to the well-known Tipper example. The results of a simulation show how this framework implements a very natural dynamic merge of inference and learning.

2 Learning the Probability Matrix

Probabilistic inference in factor graphs via message propagation is a relatively mature technique, at least in graphs with no cycles, when the conditional probability functions that make up the model are known [12]. A much harder problem is learning the model parameters on line, i.e. performing inference and learning at the same time. To focus on the specifics of this issue we start with the simplest (non trivial) factor graph of Figure 1 that models N independent realizations of two random variables $X \in \mathcal{X} = \{\xi_1, ..., \xi_d\}$ and $Y \in \mathcal{Y} = \{\eta_1, ..., \eta_m\}$. The variables are discrete and take values in the two alphabets $\mathcal{X}$ and $\mathcal{Y}$ and are related via the unknown conditional probability matrix

$$P(Y|X\Theta) = \begin{pmatrix} p(\eta_1|\xi_1) \; ... \; p(\eta_m|\xi_1) \\ p(\eta_1|\xi_2) \; ... \; p(\eta_m|\xi_2) \\ . \quad ... \quad . \\ p(\eta_1|\xi_d) \; ... \; p(\eta_m|\xi_d) \end{pmatrix} = \Theta = \begin{pmatrix} \Theta_{11} \; ... \; \Theta_{1m} \\ \Theta_{21} \; ... \; \Theta_{2m} \\ . \quad ... \quad . \\ \Theta_{d1} \; ... \; \Theta_{dm} \end{pmatrix}, \qquad (1)$$

with $0 \leq \Theta_{ij} \leq 1$, $i = 1, ..., d$, $j = 1, ..., m$; $\sum_{j=1}^{m} \Theta_{ij} = 1$, $i = 1, ..., d$. The unknown parameters make up the matrix $\Theta \in \mathcal{T}$, where $\mathcal{T}$ denotes the set of all $d \times m$ stochastic matrices. Since the structure of Figure 1 may be part of a more complex network, we assume that information on $X[n]$ and $Y[n]$ is available in

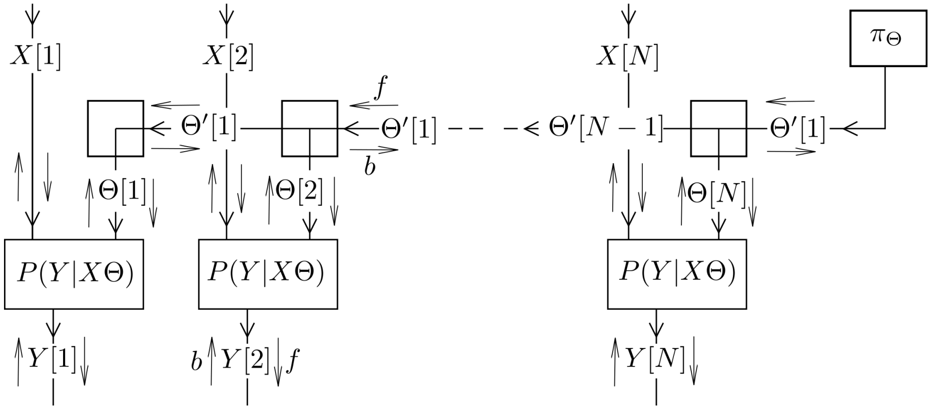

Fig. 1. The factor graph for N independent realizations of $(X[n], Y[n])$

soft form via forward and backward distributions $f_{X[n]}(x)$, $b_{X[n]}(x)$, $f_{Y[n]}(y)$ and $b_{Y[n]}(y)$, with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Also information about matrix Θ is carried by forward and backward messages $f_{\Theta[n]}(\theta)$ and $b_{\Theta[n]}(\theta)$ which are matrix functions. These messages are related to each other via marginalization as

$f_{Y[n]}(y) \propto \int_{\theta \in \mathcal{T}} \sum_{x \in \mathcal{X}} P(y|x\theta) f_{X[n]}(x) f_{\Theta[n]}(\theta) d\theta$;
$b_{X[n]}(x) \propto \int_{\theta \in \mathcal{T}} \sum_{y \in \mathcal{Y}} P(y|x\theta) b_{Y[n]}(y) f_{\Theta[n]}(\theta) d\theta$;
$b_{\Theta[n]}(\theta) \propto \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(y|x\theta) b_{Y[n]}(y) f_{X[n]}(x)$.

As usual in factor graphs, the notation $\propto$ means that the expressions are distributions except for proper normalization. The complete model is hybrid because $X[n]$ and $Y[n]$ are discrete and Θ is continuous and multi-dimensional. In a more compact matrix representation, forward and backward messages for $X[n]$ and $Y[n]$ are the column vectors

$\mathbf{f}_{X[n]} = (f_{X[n]}(\xi_1), ..., f_{X[n]}(\xi_d))^T$; $\mathbf{b}_{X[n]} = (b_{X[n]}(\xi_1), ..., b_{X[n]}(\xi_d))^T$;
$\mathbf{f}_{Y[n]} = (f_{Y[n]}(\eta_1), ..., f_{Y[n]}(\eta_m))^T$; $\mathbf{b}_{Y[n]} = (b_{Y[n]}(\eta_1), ..., b_{Y[n]}(\eta_m))^T$.

Therefore we can write

$\mathbf{f}_{Y[n]} \propto \int_{\theta \in \mathcal{T}} \theta^T \mathbf{f}_{X[n]} f_{\Theta[n]}(\theta) d\theta = F_{\theta[n]}^T \mathbf{f}_{X[n]}$;
$\mathbf{b}_{X[n]} \propto \int_{\theta \in \mathcal{T}} \theta \mathbf{b}_{Y[n]} f_{\Theta[n]}(\theta) d\theta = F_{\theta[n]} \mathbf{b}_{Y[n]}$,

where $F_{\theta[n]} = \int_{\theta \in \mathcal{T}} \theta f_{\Theta[n]}(\theta) d\theta$ is the mean forward matrix for $\Theta[n]$. The backward message for $\Theta[n]$ is the matrix function

$$b_{\Theta[n]}(\theta) \propto \mathbf{f}_{X[n]}^T \theta \mathbf{b}_{Y[n]} = \mathbf{f}_{X[n]}^T \begin{pmatrix} \theta_{11} & ... & \theta_{1m} \\ \theta_{21} & ... & \theta_{2m} \\ . & ... & . \\ \theta_{d1} & ... & \theta_{dm} \end{pmatrix} \mathbf{b}_{Y[n]}. \tag{2}$$

Messages for $\Theta[n]$ and $\Theta'[n]$ in the other branches are formally the result of the product rule $f_{\Theta[n]}(\theta) \propto f_{\Theta'[n]}(\theta) b_{\Theta'[n-1]}(\theta)$; $b_{\Theta'[n]}(\theta) \propto b_{\Theta[n]}(\theta) b_{\Theta'[n-1]}(\theta)$; $f_{\Theta'[n]}(\theta) \propto b_{\Theta[n+1]}(\theta) f_{\Theta'[n+1]}(\theta)$. Each message is a product of the type

$$\mu_\Theta(\theta) \propto \prod_l \mathbf{f}_{X[l]}^T \begin{pmatrix} \theta_{11} & ... & \theta_{1m} \\ \theta_{21} & ... & \theta_{2m} \\ . & ... & . \\ \theta_{d1} & ... & \theta_{dm} \end{pmatrix} \mathbf{b}_{Y[l]} = \prod_l \sum_{i=1}^{d} \sum_{j=1}^{m} b_{Y[l]}(\eta_j) f_{X[l]}(\xi_i) \theta_{ij} \tag{3}$$

If variables $X[n]$ and $Y[n]$ of block n are *instantiated*, i.e. forward and backward messages are delta functions, $f_{X[n]}(x) = \delta(x - \xi_i)$, $b_{Y[n]}(x) = \delta(y - \eta_j)$, backward information from block n is simply $b_{\Theta[n]}(\theta) \propto \theta_{ij}$. If also all variables from all n are instantiated, information exchanged among the blocks (except possibly for the prior on Θ) are exactly products of Dirichlet distributions

$$\mu_\Theta(\theta) \propto \prod_{i=1}^{d} \prod_{j=1}^{m} \theta_{ij}^{n_{ij}} \propto \prod_{i=1}^{d} Dir(\theta_{i1}, ..., \theta_{im}; n_{i1} + 1, ..., n_{im} + 1), \tag{4}$$

where n_{ij} are the integer numbers that represent the cumulative counts of the occurrences of pair (i, j) (*hard scores*). Unfortunately, in the general case we are interested in with forward and backward messages carrying soft information, expression (3) becomes intractable. Hence we resort to a variational approximation [5][3][18] for $b_{\Theta[n]}(\theta)$ that gives

$$\begin{array}{l} b^V_{\Theta[n]}(\theta) \propto e^{\sum_{i=1}^d \sum_{j=1}^m b_{Y[n]}(\eta_j) f_{X[n]}(\xi_i) \log \theta_{ij}} = \prod_{i=1}^d \prod_{j=1}^m \theta_{ij}^{b_{Y[n]}(\eta_j) f_{X[n]}(\xi_i)} \\ \propto \prod_{i=1}^d Dir(\theta_{i1}, ..., \theta_{im}; f_{X[n]}(\xi_i) b_{Y[n]}(\eta_1) + 1, ..., f_{X[n]}(\xi_i) b_{Y[n]}(\eta_m) + 1), \end{array} \tag{5}$$

which is again the product of d Dirichlet distributions. This is particularly interesting because the Dirichlet distribution, sometimes used as an assumption [7][19], is exactly the variational approximation. Assuming that also the prior distribution π_Θ is a product of Dirichlet functions
$\pi_\Theta \propto \prod_{i=1}^d Dir(\theta_{i1}, ..., \theta_{im}; \alpha_{i1} + 1, ..., \alpha_{im} + 1)$.

A generic message in the upper branches has the form

$$\begin{array}{l} \mu_\Theta \propto \prod_{i=1}^d Dir(\theta_{i1}, ..., \theta_{im}; \\ \quad \alpha_{i1} + \sum_l f_{X[l]}(\xi_i) b_{Y[l]}(\eta_1) + 1, ..., \alpha_{im} + \sum_l f_{X[l]}(\xi_i) b_{Y[l]}(\eta_m) + 1). \end{array} \tag{6}$$

A priori knowledge about the rule that maps X into Y can also be easily included in the coefficients of π_Θ. The exponential form for the variational approximation suggests that matrix variables $\Theta[n]$ and $\Theta'[n]$ could be replaced with *soft score* matrix variables $O[n]$ and $O'[n]$. Backward message from block n becomes matrix $b_{O[n]} = \mathbf{f}_{X[n]} \mathbf{b}^T_{Y[n]}$. Also all messages in the upper branches become $d \times m$ matrices with combination rules $f_{O[n]} = f_{O'[n]} + b_{O'[n-1]}$; $b_{O'[n]} = b_{O[n]} + b_{O'[n-1]}$; $f_{O'[n]} = b_{O[n+1]} + f_{O'[n+1]}$. Forward and backward messages for $Y[n]$ and $X[n]$ are respectively $\mathbf{f}_{Y[n]} \propto F^T_{O[n]} \mathbf{f}_{X[n]}$; $\mathbf{b}_{X[n]} \propto F_{O[n]} \mathbf{b}_{Y[n]}$,, where $F_{O[n]}$ is the row-normalized version of $f_{O[n]}$. Note that these propagation rules represent the learning steps for Θ as inference and learning happen at the same time. Recall that the various stages in the graph represent time-unfolded versions of the same block. Mode details and proofs will be reported in a longer paper.

3 Soft Quantization

Manipulation of discrete quantities in machine learning, also when the problem involves continuous variables, may be particularly handy, because a priori qualitative information can be more easily injected into the system. Fuzzy methods [17] have shown great success in merging soft knowledge with hard functions especially in control [9]. In [15] we have shown how the use entropic priors in the Bayesian framework allowing the introduction of soft membership information in a way that is consistent within standard probability theory. This is a crucial step to allow soft quantization and coherent use of probability propagation for inference and learning in systems that contain both continuous and discrete variables.

Figure 2 shows a quantization scheme for a continuous variable S_a. All the likelihoods are triangular, complementary and centered on the M nodes $\xi_1, ..., \xi_M$. Denoting the triangular function on a, b, c with $\Lambda(s_a; a, b, c)$, the M pdfs are

$$\begin{aligned}&\{\tfrac{2}{\xi_2-\xi_1}\Lambda(s_a;\xi_1,\xi_1,\xi_2), \tfrac{2}{\xi_3-\xi_1}\Lambda(s_a;\xi_1,\xi_2,\xi_3),\\ &..., \tfrac{2}{\xi_M-\xi_{M-2}}\Lambda(s_a;\xi_{M-2},\xi_{M-1},\xi_M), \tfrac{2}{\xi_M-\xi_{M-1}}\Lambda(s_a;\xi_{M-1},\xi_M,\xi_M)\},\end{aligned} \tag{7}$$

and are shown in Figure 2(a). The differential entropy [4] of $\Lambda(s_a; a, b, c)$ is easily computed to be $h(S_a) = \frac{1}{2} + \log\frac{c-a}{2}$. With entropic priors $\pi_i \propto e^{h(S_a|i)}$ [15], the prior-likelihood products, become equivalent to a set of functions with same height as in Figure 2(b). We recall that entropic priors are the distribution that maximize the joint entropy $H(S_a, S)$ for fixed likelihoods $(p_{S_a}(s_a|1), ..., p_{S_a}(s_a|M))$ [15]. The node distribution can be chosen according to the data points density, but the complementarity of the likelihoods guaratees that no information is lost after soft quantization. Figure 2(b) shows also how this kind of soft quantization can be drawn as a generative factor graph model that can be inserted into a larger factor graph. The backward message for S_a is a data point $b_{S_a}(s_a) = \delta(s_a - s_0)$. The backward message for S^1 in vector notation is $\mathbf{b}_{S^1} = (p_{S_a}(s_0|1), ..., p_{S_a}(s_0|M))^T$ that after combination with entropic priors becomes $\mathbf{f}_{S^2} = (p_{S_a}(s_0|1)\pi_1, ..., p_{S_a}(s_0|M)\pi_M)^T$. The soft quantization model satisfies a property of perfect recostruction because if $\mathbf{b}_{S^2} = \mathbf{b}_{S^1}$, we have $\mathbf{f}_{S^1} = \mathbf{f}_{S^2}$ and $f_{S_a}(s_a) = \delta(s_a - \mathbf{f}^T_{S^1}(\xi_1, ..., \xi_M)^T) = \delta(s_a - s_0)$ (lossless dequantization). More details about soft quantization with entropic priors will be reported in a longer paper elsewhere.

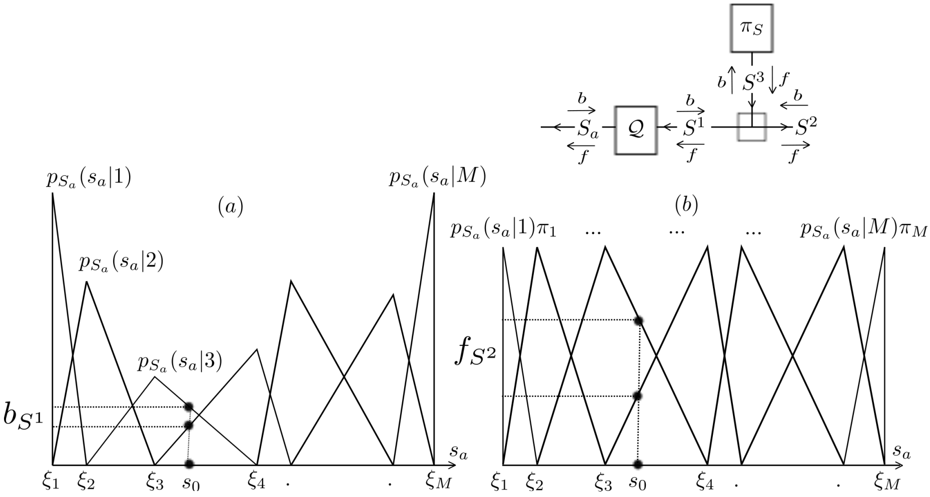

Fig. 2. Soft quantization on nodes $\{\xi_1, ..., \xi_M\}$. (a) The triangular likelihoods; (b) The entropic priors-likelihoods products.

4 The Tipper Example

In this paper we report some experiments with the variational learning rules of Section 2 and with the soft quantization scheme of Section 3 on the well-known "Tipper" example. In this problems there are three continuous variables: S_a (Service), F_a (Food) and T_a (Tip). The Tipper example, often used as a teaching example in control classes (there is a Matlab demo available in the Fuzzy Control Toolbox), is a typical case of mapping between two input variables (Service and Food) and a final one (Tip). In the fuzzy framework is also very easy to include soft rules and various design constraints. Our objective is here to traslate this typical approach into a probabilistically-consistent Bayesian framework. The underlying factor graphs shown in Figure 4, in which messages travel back and forth, allows simultaneous inference and learning with inputs and outputs that become essentially indistinguishable.

Even though a priori soft-logic rules can be easily included as contraints in the prior block π_Θ, we have assumed here no prior knowledge about the variables S_a, F_a and T_a. We have simply presented 50 realizations of the triplet as f_{S_a}, f_{F_a} and f_{T_a} and let the system learn (simulations with combinations of soft rules and examples will be reported elsewhere). The triplets were obtained from a blind run of the Matlab demo. Forward and backward messages carry information in various parts of the system and inferences can also be made backward on Service and/or Food from Tip.

The graph structure assumes that variables Service and Food are mutually independent and that the $N = 50$ realizations are also statistically independent. The three analog variables S_a, F_a and T_a are soft quantized from ranges $[0-10][0-10][5-25]$ with $M = 6$ uniformly spaced nodes each into the three discrete variables S, F and T. Entropic priors are imposed in π_S, π_F and π_T. The 36×6 matrix of conditional probabilities $P(T|SF\Theta)$ is learned via message propagations with the variational algorithm described in Section 2. The simulations let the messages propagate 300 steps which is enough to cover the graph diameter. The graph is clearly a tree and convergence is guaranteed. Figure 4 shows the comparison of forward and backward information at each stage n. The thre plots show the comparison of the actual value of each variable, as carried by the backward input message, with the value provided by the rest of the system, as carried by the forward output message that uses all the other inputs after learning and propagation. Note that learning and inference is all done at the same time since information about the parameter θ are also carried by travelling messages. The simulation is self-contained and implements our best use of the data because the inference, say on $T_a[n]$, is based on all the examples except the one on $T_a[n]$. This is because $f_{\Theta[n]}$ does not contain information coming from $b_{\Theta[n]}$. Hence each stage n uses a slightly different estimate for $P(T|SF\Theta)$ because the nth examples is automatically excluded. Therefore in inferring $T_a[n]$, values of $S_a[n]$ and $F_a[n]$ are used for inference, but not for learning. The same considerations apply to inferences on $S_a[n]$ and $F_a[n]$.

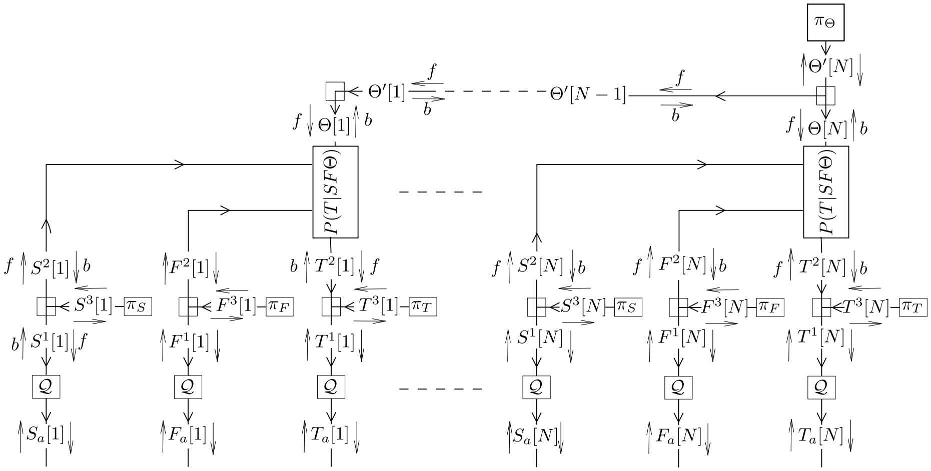

Fig. 3. The factor graph for the Tipper example

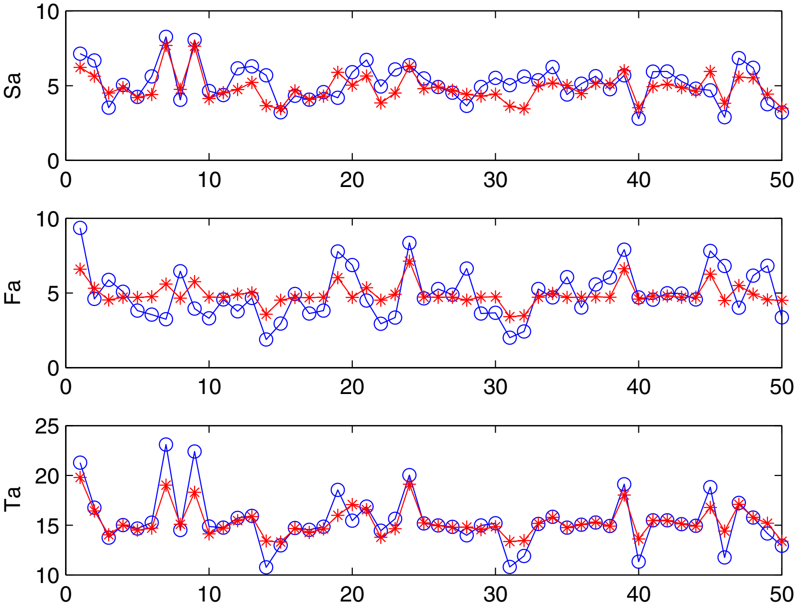

Fig. 4. Comparison of forward (inference) (*) and backward (true) (o) values for the three variables in the Tipper example

Conclusions

In this work we have reported partial results for a successful Bayesian paradigm that implements via message propagation on a factor graph simultanaous inference and learning. By means of an example, we have also proposed a quantization scheme, that via the introduction of entropic priors, allows us to build a probabilistically-consistent graph that can adapted with belief propagation. More work will be devoted to further understanding of the adaptation rules and on the inclusion of soft-logic contraints.

References

1. Graphical models emerge. new connections betweeen machine learning and signal processing. Signal Processing Magazine, 27(6) (2010)
2. Beal, M.J.: Variational algorithms for approximate bayesian inference. Ph.D. thesis, University of London (2003)
3. Beal, M.J., Ghahramani, Z.: Variational bayesian learning of directed graphical models with hidden variables. Bayesian Analysis 1, 1–44 (2004)
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley (2006)
5. Dauwels, J.: On variational message passing in factor graphs. In: ISIT2007, Nice, France, June 24-June 29 (2007)
6. Ghahramani, Z.: Unsupervised Learning. Springer (2004)
7. Heckerman, D.: A tutorial on learning with bayesian networks. Tech. Rep. MSR-TR-95-06, Microsoft Research (1996); March 1995 (Revised November 1996)
8. Dauwels, J., Eckford, A., Loeliger, S.K., Expectation, H.A.: xpectation maximization as message passing–part i: Principles and gaussian messages. arXiv:0910, 1–14 (2009); Submitted to IEEE Tr. on Information Theory
9. Jantzen, J.: Foundations of Fuzzy Control. Wiley (2007)
10. Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press (2003)
11. Loeliger, H.A.: An introduction to factor graphs. IEEE Signal Processing Magazine 21(1), 28–41 (2004)
12. Loeliger, H.A., Dauwels, J., Hu, J., Korl, S., Ping, L., Kschischang, F.: The factor graph approach to model-based signal processing. Proceedings of the IEEE 95(6), 1295–1322 (2007)
13. Choi, M.J., Tan, V.Y.F., Anandkumar, A., Willsky, A.S.: Learning latent tree graphical models. Journal of Machine Learning Research 12, 1771–1812 (2011)
14. Palmieri, F.A.N., Ciuonzo, D.: Entropic priors for short-term stochastic process classification. In: 14th Int. Conf. on Information Fusion, Chicago, IL (2011)
15. Palmieri, F.A.N., Ciuonzo, D.: Objective priors from maximum entropy in data classification. In: Information Fusion (2012), doi:10.1016/j.inffus.2012.01.012
16. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
17. Novak, V., Perfilieva, I., Mockor, J.: Mathematical Principles of Fuzzy Logic. Kluwer Academic Press (1999)
18. Winn, J., Bishop, C.M.: Variational message passing. Journal of Machine Learning Research 6, 661–694 (2005)
19. Winn, J.M.: Variational message passing and its applications. Ph.D. thesis, University of Cambridge (2004)