Unsupervised (parameter) learning for MRFs on bipartite graphs

Boris Flach
http://cmp.felk.cvut.cz/~flachbor
Tomas Sixta
http://cmp.felk.cvut.cz/~sixtatom

Center for Machine Perception
Czech Technical University
Prague, Czech Republic

This paper considers unsupervised (parameter) learning for general MRFs on bipartite graphs. That means that we assume training samples which consist of i.i.d. realisations of the field of visible variables only. The variables indexed by the vertices of the other graph part are assumed to be latent. The corresponding learning task is non-trivial because the (log) likelihood is a non-concave function of the model parameters, and, what is worse, its gradient is not tractable. A common approach is to calculate an approximation of the gradient by applying a stochastic gradient method known as "Persistent Contrastive Divergence" [5]. Another option, discussed in [3], is to marginalise over the latent variables (what can be done up to the unknown partition sum) and to maximise the pseudo-likelihood for the resulting higher order model directly. Notice however, that the resulting objective function is then no longer concave.

The main contribution of this paper is to introduce an alternative learning approach – a modified EM-algorithm with pseudo-likelihood estimator in the M-step, which is tractable on account of the bipartiteness of the model graph. In principle, such a modified EM-algorithm can be applied for parameter learning of arbitrary MRFs [6]. The resulting algorithm will however remain to be intractable, because so is the computation of the posterior pairwise marginal probabilities in the E-step. It is the bipartiteness of the graph, which ensures that the E-step *and* the M-step of the EM-algorithm are both tractable, if the maximum likelihood estimator in the M-step is replaced by the pseudo-likelihood estimator.

MRFs on bipartite graphs can be described as follows. Let (V,E) be an undirected bipartite graph and V_1, V_2 denote its parts. Let X be a collection of K_1-valued random variables indexed by vertices of V_1. That is, $X = \{X_i \mid i \in V_1\}$, where each X_i is a K_1-valued random variable. Similarly, Y denotes a collection of K_2-valued random variables indexed by vertices of the second part V_2. Both co-domains K_1 and K_2 are assumed finite. We denote realisations of the random field (X,Y) by (x,y), *i.e.* $x\colon V_1 \to K_1$ and $y\colon V_2 \to K_2$.

The joint p.d. of an MRF on (V,E) can be written as an exponential family (assuming strictly positive probability mass)

$$p_u(x,y) = \frac{1}{Z(u)} \exp \sum_{ij\in E} \big\langle \boldsymbol{\varphi}(x_i,y_j), \boldsymbol{u}_{ij} \big\rangle, \tag{1}$$

where $\boldsymbol{\varphi}\colon K_1 \times K_2 \to \mathbb{R}^{|K_1||K_2|}$ designates the vector valued indicator mapping $\varphi_{lm}(k,k') = \delta_{lk}\delta_{mk'}$ and $u = \{\boldsymbol{u}_{ij} \mid ij \in E\}$ denotes the set of model parameters. This model class includes Restricted Boltzmann Machines [2] which are often used in the context of deep learning [1].

We assume from here on that the variables X_i, $i \in V_1$ are visible, whereas the variables Y_j, $j \in V_2$ are latent and consider the task of parameter estimation given an i.i.d. sample $\mathcal{T}_\ell$ of ℓ realisations of the field X. It is assumed that the realisations were generated by $p_u(x) = \sum_{y\in\mathcal{Y}} p_u(x,y)$ with unknown u. If the maximum likelihood estimator is used, the task is

$$\frac{1}{\ell} \sum_{x\in\mathcal{T}_\ell} \log \sum_{y\in\mathcal{Y}} p_u(x,y) \to \max_u, \tag{2}$$

where $\mathcal{Y}$ denotes the set of all possible realisations of the field Y. A common method to maximize the likelihood (2) in the presence of incomplete data is the EM-algorithm.
E-step: Calculate posterior probabilities

$$\beta^{(t)}(y \mid x) := p_{u^{(t)}}(y \mid x) \tag{3}$$

for each realisation $x \in \mathcal{T}_\ell$ using the current parameter estimate $u^{(t)}$. This task is feasible for the considered model class, because the conditional p.d. $p_u(y \mid x)$ factorises.
M-step: Given the current $\beta^{(t)}$ maximise the log-likelihood for complete information

$$L_c(u) = \frac{1}{\ell} \sum_{x\in\mathcal{T}_\ell} \sum_{y\in\mathcal{Y}} \beta(y \mid x) \log p_u(x,y) \to \max_u. \tag{4}$$

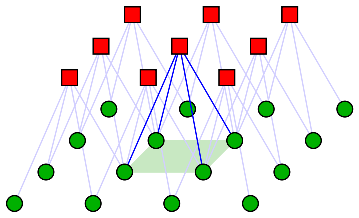
Figure 1: MRF on a translational invariant bipartite graph. Visible variables depicted as green circles, latent variables depicted as red squares. Edges and receptive field are highlighted for one of the latent variables.

This objective function is concave in u but computing the components of the gradient requires to compute pairwise marginal statistics of the model $p_u(x,y)$ and is therefore not tractable.

Following the interpretation given by one of the authors of the EM-algorithm [4], the task to be solved in the M-step is itself a (parameter) learning task, now in presence of complete data. The model parameters u must be estimated given the "observed" distribution $p^*(x,y) = \beta(y \mid x)p^*(x)$, where $p^*(x)$ is the empirical distribution associated with the sample $\mathcal{T}_\ell$. This implies that i.i.d. samples from p^* can be easily generated. The key idea is therefore to replace the maximum likelihood estimator in the M-step by any consistent *and* tractable estimator. A reasonable choice is the pseudo-likelihood estimator.

Let us denote by $\mathcal{T}^*$ an i.i.d. sample of realisations (x,y) generated from $p^*(x,y)$. The pseudo-likelihood estimator for MRFs on bipartite graphs reads

$$L_p(u) = \sum_{(x,y)\in\mathcal{T}^*} \Big[\log p_u(y \mid x) + \log p_u(x \mid y)\Big] \to \max_u. \tag{5}$$

This objective function is concave and has a tractable gradient. Optimization task in the M-step then reads to maximise the pseudo-likelihood (5) e.g. by using a gradient ascend algorithm.

In the rest of the paper we discuss implementation details of the proposed algorithm and provide experimental comparison with the PCD. We show, that despite the same per iteration time complexity our algorithm converges faster (by an order of magnitude) and more stable. On the other hand, we have no proof that the sequence of likelihood values $L(u^{(t)})$ is increasing. This should be true in the limit of an infinite training sample because the pseudo-likelihood estimator is known to be consistent.

[1] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[2] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

[3] Benjamin Marlin, Kevin Swersky, Bo Chen, and Nando de Freitas. Inductive principles for restricted Boltzmann machine learning. *Journal of Machine Learning Research*, 9:509–516, 2010.

[4] Michail I. Schlesinger. The interaction of learning and self-organization in pattern recognition. *Cybernetics and Systems Analysis*, 4(2):66–71, 1968.

[5] T. Tieleman. Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM New York, NY, USA, 2008.

[6] Muneki Yasuda, Junya Tannai, and Kazuyuki Tanaka. Learning algorithm for boltzmann machines using max-product algorithm and pseudo-likelihood. *Interdisciplinary Information Sciences*, 18(1):55–63, 2012.