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

Abstract

We consider unsupervised (parameter) learning for general Markov random fields on bipartite graphs. This model class includes Restricted Boltzmann Machines. We show that besides the widely used stochastic gradient approximation (a.k.a. Persistent Contrastive Divergence) there is an alternative learning approach – a modified EM algorithm which is tractable because of the bipartiteness of the model graph. We compare the resulting double loop algorithm and the PCD learning experimentally and show that the former converges faster and more stable than the latter.

1 Introduction

Markov random fields (MRF) provide a good basis for modelling joint distributions of collections (fields) of interdependent random variables. In order to express complex interdependencies between them, it is necessary to introduce factors of higher order (arity) which, however, are complicated to handle if considered non-parametric. A common compromise is to restrict the arity of the factors, but to introduce additional latent variables. Marginalising over the latent variables then leads to a model with higher arity factors for the remaining, "visible" variables. Such and similar approaches have been used e.g. in computer vision for several models: fields of experts [8], regression tree fields [5] to name a few.

One of the most simple model classes of such type are MRFs on bipartite graphs, where vertices of the first part index "visible" random variables and vertices of the second part index latent variables. A subclass of such models is well known as Restricted Boltzmann Machines (RBM) [4]. They are often used as building blocks of hierarchical models in the context of "deep learning" (see [1] for a review).

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 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" [13, 14]. Another option, discussed in [7], 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 for the mentioned model class – 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 [15]. 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.

We will fix the notations, the model class and the learning task in the next section and introduce the tractable modification of the EM algorithm in section 3. An experimental comparison of the resulting double loop algorithm with the PCD method given in section 4, shows that the former converges faster (by an order of magnitude) and more stable than the latter.

2 The model and the learning task

2.1 The model class

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.*

$$x\colon V_1 \to K_1, \quad y\colon V_2 \to K_2.$$

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 [4] which are often used in the context of deep learning [1]. An RBM in its narrow sense assumes that the co-domains of both groups of random variables are binary $|K_1| = |K_2| = 2$ and the bipartite model graph is complete.

Despite the fact that the considered model class has pairwise factors only, it can be used to model higher order factors in the following way. If the variables X_i, $i \in V_1$ are considered as "visible" and the variables Y_j, $j \in V_2$ as latent, then, by marginalising over the field Y, we get a Gibbs Random Field with higher order factors for the field X.

Notice that due to the bipartiteness of the graph both conditional p.d.s $p_u(y \mid x)$ and $p_u(x \mid y)$ factorise

$$p_u(y \mid x) = \prod_{j\in V_2} p_u(y_j \mid x_{\mathcal{N}_j}), \tag{2}$$

where $\mathcal{N}_j = \{i \in V_1 \mid ij \in E\}$ denotes the neighbourhood of the vertex $j \in V_2$.

2.2 The learning task

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

$$\frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \log \sum_{y \in \mathcal{Y}} p_u(x,y) \to \max_u, \tag{3}$$

where $\mathcal{Y}$ denotes the set of all possible realisations of the field Y. Substituting the model class (1), the task reads

$$L(u) = \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \log \sum_{y \in \mathcal{Y}} \exp \sum_{ij \in E} \big\langle \boldsymbol{\varphi}(x_i, y_j), \boldsymbol{u}_{ij} \big\rangle - \log Z(u) \to \max_u \tag{4}$$

where

$$Z(u) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \exp \sum_{ij \in E} \big\langle \boldsymbol{\varphi}(x_i, y_j), \boldsymbol{u}_{ij} \big\rangle \tag{5}$$

denotes the partition sum. It is easily seen that both terms in (4) are convex functions of u. The log-likelihood $L(u)$ is therefore a difference of convex functions.

3 Algorithms for the learning task

3.1 Discussion of existing methods

The gradient of the log-likelihood is easy to derive

$$\nabla_{\boldsymbol{u}_{ij}} L(u) = \frac{1}{\ell} \sum_{x \in \mathcal{T}_\ell} \mathbb{E}_u(\boldsymbol{\Phi}_{ij} | X = x) - \mathbb{E}_u(\boldsymbol{\Phi}_{ij}), \tag{6}$$

where $\boldsymbol{\Phi}_{ij}$ denotes the random variable $\boldsymbol{\Phi}_{ij}(X,Y) = \boldsymbol{\varphi}(X_i, Y_j)$, $ij \in E$. The first term in (6) is tractable because the conditional p.d. $p_u(y \mid x)$ factorises, which makes the computation of the conditional expectations tractable and because the sum over the elements of the learning sample $\mathcal{T}_\ell$ is tractable. The second term is, on the contrary, not tractable – it requires to compute pairwise marginal probabilities $p_u(x_i, y_j)$. It is well known, that calculating the marginals for an MRF is #P hard [2]. Therefore, one has to rely on approximate algorithms. Let us shortly discuss possible options.

Variational methods like belief propagation or other message passing algorithms fail to estimate pairwise marginal statistics even approximately [3]. This can be explained by the following argument. All these methods approximate the pairwise log-marginals by

$$\log p(x_i = k, y_j = k') \sim a_i(k) + u_{ij}(k,k') + b_j(k'), \tag{7}$$

i.e. as being equal to $\boldsymbol{u}_{ij}$ up to a modular function. While this is true for trees, it is wrong for general graphs because correlations caused by loops are ignored.

Another option for estimating the required marginals is Gibbs sampling. However, Gibbs sampling is very slow if applied correctly [12]. To generate just one realisation (x,y), it is often necessary to run thousands of iterations of the sampler.

A third option is a *stochastic gradient* method which is often used in the context of RBMs and is designated as Persistent Contrastive Divergence [13, 14]. PCD keeps a realisation $(x^{(t)},y^{(t)})$ at each iteration t. The current model estimate $u^{(t)}$ is used to re-sample the realisation $(x^{(t+1)},y^{(t+1)})$ The new realisation is then used to estimate the second term of the gradient, simply by replacing the expectation of $\boldsymbol{\Phi}_{ij}$ by its realisation $\boldsymbol{\varphi}(x_i,y_j)$. Finally, a new model estimate $u^{(t+1)}$ is obtained by applying a gradient step. Clearly, there are no guarantees for convergence to the global optimum because the objective function is not concave and the true gradient is replaced by an approximation.

We may try to avoid to deal with $L(u)$ directly by applying the EM algorithm. An iteration of it reads as follows.
E-step: Calculate posterior probabilities

$$\beta^{(t)}(y \mid x) := p_{u^{(t)}}(y \mid x) \tag{8}$$

for each realisation $x \in \mathcal{T}_\ell$ using the current parameter estimate $u^{(t)}$. This task is feasible for the considered model class (see (2)).
M-step: Given the current $\beta^{(t)}$ maximise the log-likelihood for complete information

$$L_c(u) = \frac{1}{\ell} \sum_{x\in\mathcal{T}_\ell} \sum_{y\in\mathcal{Y}} \beta(y \mid x) \log p_u(x,y) \to \max_u . \tag{9}$$

Let us denote by p^* the distribution $p^*(x,y) = \beta(y \mid x)p^*(x)$, where $p^*(x)$ is the empirical distribution associated with the sample $\mathcal{T}_\ell$. Substituting the model (1), the objective function in the M-step can be written as

$$L_c(u) = \frac{1}{\ell} \sum_{ij\in E} \big\langle \mathbb{E}_{p^*}(\boldsymbol{\Phi}_{ij}), \boldsymbol{u}_{ij} \big\rangle - \log Z(u). \tag{10}$$

It is concave in u, but, again, the problem is the gradient of the second term (the logarithm of the partition sum Z). Computing its components requires to compute pairwise marginal statistics of the model $p_u(x,y)$ and is therefore not tractable.

3.2 A modified EM algorithm

Following the interpretation given by one of the authors of the EM-algorithm [10], the task to be solved in each 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 $p^*(x,y)$. As we have seen, this task is still not tractable for the considered class of MRFs. On the other hand, the definition of p^* 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

$$L_p(u) = \sum_{(x,y)\in\mathcal{T}^*} \Big[\log p_u(y \mid x) + \log p_u(x \mid y) \Big] \to \max_u . \tag{11}$$

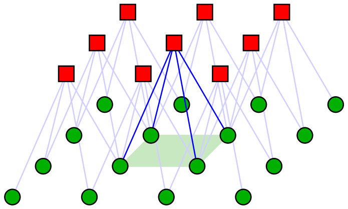

Figure 1: MRF on a translational invariant bipartite graph. Visible variables depicted as green circles, latent variables depicted as red squares. Edges and receptive field are highlighted for one of the latent variables.

The objective function is concave and has a tractable gradient

$$\nabla_{\boldsymbol{u}_{ij}} L_p(u) = \sum_{(x,y)\in\mathcal{T}^*} \Big[2\boldsymbol{\varphi}(x_i,y_j) - \mathbb{E}_u(\boldsymbol{\Phi}_{ij}|X=x) - \mathbb{E}_u(\boldsymbol{\Phi}_{ij}|Y=y) \Big]. \tag{12}$$

Summarising, each iteration of the modified EM algorithm reads as follows
E-step: Calculate posterior probabilities

$$\beta^{(t)}(y \mid x) := p_{u^{(t)}}(y \mid x) \tag{13}$$

for each realisation $x \in \mathcal{T}_\ell$ using the current parameter estimate $u^{(t)}$. Sample one (or several) realisations y for each $x \in \mathcal{T}_\ell$. These data define the current sample $\mathcal{T}^*$ for the M-step.
M-step: Maximise the pseudo-likelihood

$$L_p(u) = \sum_{(x,y)\in\mathcal{T}^*} \Big[\log p_u(y \mid x) + \log p_u(x \mid y) \Big] \tag{14}$$

e.g. by using a gradient ascend algorithm. Set $u^{(t+1)}$ to be equal to the maximiser.

It remains to discuss the choice for the initial model parameters $u^{(0)}$. The simplest option is to choose them randomly in the vicinity of the origin. Yet there is a better option for MRFs on bipartite graphs. Let us consider the sub-graph defined by a vertex $j \in V_2$ and its neighbours $\mathcal{N}_j \subset V_1$ and the random variables Y_j, X_i, $i \in \mathcal{N}_j$. Taken alone, they define a naive Bayes model. The parameters of such a model can be learned by a standard EM-algorithm. Applying it for each of the sub-models separately, gives a good initialisation for the model parameters.

In summary, the resulting double loop algorithm is easy to implement and has the same per iteration time complexity as PCD. 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. However, there is no such guarantee for finite training samples. We will compare the proposed algorithm with PCD for direct likelihood maximisation in the experimental section.

4 Experiments

We aim to apply the discussed type of MRFs for shape modelling. By this we mean to model simple shapes and spatial relations (like “above, ”inside“, etc.) for segments. Bearing in

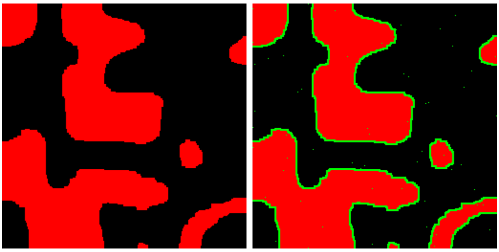

Figure 2: A realisation (x,y) generated by the P9 model.

mind such applications, we make the following assumptions for all presented experiments. The vertex sets V_1 and V_2 are congruent subsets of $\mathbb{Z}^2$ and the values of the random variables x_i represent segment labels. The graph structure is translation invariant, *i.e.*, $\mathcal{N}_{j+a} = \mathcal{N}_j + a$ for all $j, a \in \mathbb{Z}^2$ such that $j, j+a \in V_2$ (see Fig. 1). We call $\mathcal{N}_j$ receptive field of the latent variable Y_j. The model parameters are translation invariant as well

$$\boldsymbol{u}_{ij} = \boldsymbol{u}_a, \quad \forall \{i,j\} \in E \text{ s.t. } i - j = a. \tag{15}$$

Please notice that the models we are using here for experiments differ from those usually used for experiments on RBMs (see *e.g.* [7]) in two respects. We use large size fields in contrast to usually used models of relatively small size. The latent variables are often considered as features for subsequent classification. Here in contrast, they are used to model complex distributions for the field X.

4.1 The Pn model

We consider the *Pn* model for binary segmentations in the first experiment. It is a generalised Potts model on cliques of size n [6]. The factors of the Gibbs Random Field associated with the cliques $\mathcal{N}_j$ are two-valued; a large value is assigned to homogeneous realisations of $X_{\mathcal{N}_j}$ with either of the two possible segment labels. A small value is assigned to all other realisations of $X_{\mathcal{N}_j}$. To express this higher order model by an MRF on a bipartite graph, we make each clique $\mathcal{N}_j$ a receptive field of a three-valued latent variable Y_j. The conditional p.d.s $p(x_{\mathcal{N}_j} \mid y_j)$ for the first two values of Y_j are non-zero only for the two homogeneous realisations $x_{\mathcal{N}_j} \equiv 0, 1$ respectively. The conditional p.d. for the third value of Y_j is uniform. A mixture of the three p.d.s corresponds to a factor of the *Pn* model.

We have implemented a *P*9 model with receptive fields of size 3×3. Fig. 2 shows a random realisation (x,y) (colour coded) generated by this model. We have generated 50 realisations of X (size 256x256) by extensive Gibbs sampling (10^4 sampling iterations per example) and used them for learning. The model was learned by the stochastic gradient method (PCD) and by the proposed modified EM-algorithm. In this experiment we were not using the "naive Bayes"-based initialisation (see sec. 3.2). We have chosen the size 512x512 for the realisation (x,y) needed for the gradient estimation in the PCD algorithm. The optimal step width for the gradient ascends were chosen empirically for each of the algorithms.

To compare the two learning algorithms, we display the $\mathbb{L}_\infty$ norm of the gradients over the iteration number in Fig. 3. The sawtooth-like shape of the curve for the modified EM-algorithm is explained as follows. The (negative) pseudo-likelihood and its gradient decrease

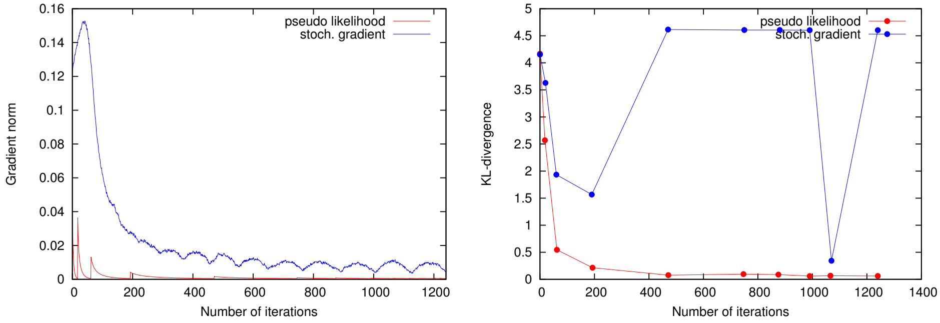

Figure 3: Comparison of stochastic gradient method and the proposed modified EM-algorithm. Left: norm of the gradient. Right: KL divergence from true marginals

in the inner loop of the algorithm (M-step). Then the y-fields are re-sampled using the new model estimate (E-step), what causes the jump in the gradient of the pseudo-likelihood.

Overall, it is clearly seen that the proposed modified EM-algorithm converges faster by an order of magnitude and much more stable than the stochastic gradient algorithm. Of course, this comparison alone does not say anything about the models learned by the respective algorithm. The objective functions are different and, moreover, the gradient of the likelihood (in the PCD algorithm) is determined approximately only.

It would not be very reasonable to compare the learned models by comparing their parameters u directly. They are not unique due to possible re-parametrisations. Moreover, models with different distributions $p_u(x,y)$ may have the same distribution $p_u(x)$. Therefore we have chosen to compare the resulting marginal distributions $p_u(x_{\mathcal{N}_j})$ for the receptive fields of size 3x3 which have 512 possible realisations. They were estimated for the true model as well as for each of the learned models by extensive sampling. Fig. 3 shows the KL-divergence between the marginals of the true model and the learned models for some iteration numbers. Again, it is clearly seen that the proposed modified EM-algorithm converges faster and much more stable than the PCD algorithm.

4.2 Cell segmentations

We consider a more complex model for the second experiment. The goal is to learn a prior model for segmenting cells in microscope images. We assume a typical segmentation to contain non-occluding cells with roughly circular shaped cytoplasm and circular shaped nuclei. Artificial segmentations of the type shown in Fig. 4 were used as training data. To learn such segmentations we have chosen a model with the following structure. The co-domain K_1 of the variables X_i has three values corresponding to the three possible segment labels – background, cytoplasm, nucleus. The co-domain K_2 of the latent variables was chosen to have five values. The receptive fields for the latter were chosen to have roughly the size of a cell, 11×11 pixels in our case. To speed up learning, we used the ”naive Bayes“-initialisation (see sec. 3.2).

Fig. 4 shows the learning curves for the modified EM-algorithm and the stochastic gradient algorithm. Again, the former converges much faster and more stable than the latter. Moreover, comparing realisations generated by the learned models (see Fig. 5), it is seen

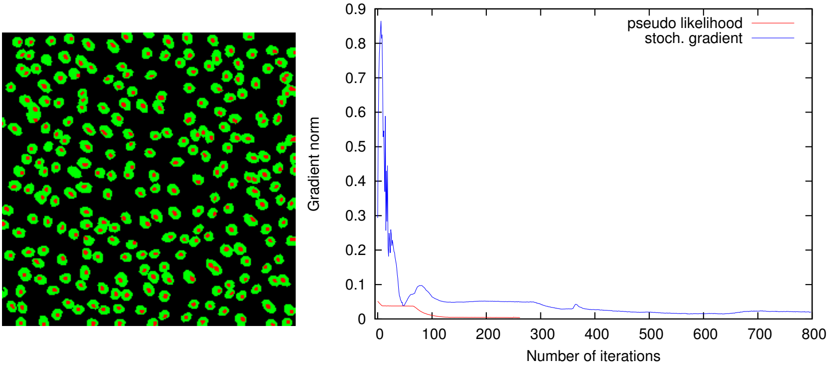

Figure 4: Left: cell segmentation (artificial). Right: Comparison of stochastic gradient method and the proposed modified EM-algorithm.

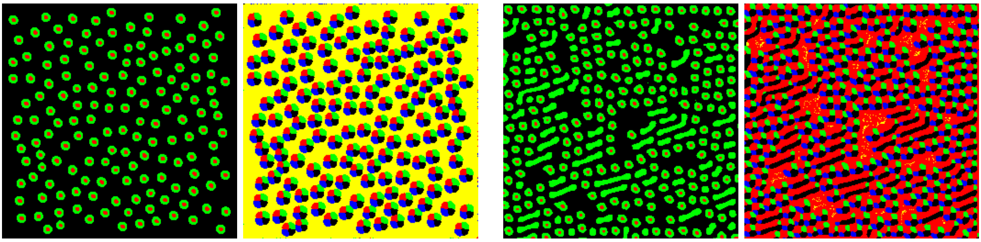

Figure 5: Realisations (x,y) randomly generated by the learned cell segmentation models. Left pair: model learned by modified EM-algorithm. Right pair: model learned by stochastic gradient algorithm. Three colours (red, green, black) were used to represent the possible values of x_i, $i \in V_1$ and four colours (red, green, black, yellow) were used to represent the possible values of the latent variables y_j, $j \in V_2$.

that the model learned by the modified EM-algorithm generates desired segmentations after 260 learning cycles. The model learned by the PCD algorithm has not yet fully ”captured“ the desired segmentations even after 800 learning cycles.

4.3 Lung segmentation

The aim of the last experiment differs from those of the previous experiments – here we want to demonstrate the usefulness of MRFs on bipartite graphs for segmentation tasks. Let us consider lung segmentation in X-ray chest radio-graphs as an example. As typical for such tasks, it is desirable to have a segmentation model which prefers e.g. smooth boundaries and simultaneously utilises a probabilistic anatomical atlas. This is easy to achieve by using models of the considered type. A translational invariant model as shown in Fig. 1 is extended by one more latent variable with edges to all pixels of the segmentation. This ”global“ latent variable realises a ”mixture“ of anatomical atlases jointly with the other latent variables, which model translational invariant local segment/boundary features.

Such a model was used as a prior model for segmenting lungs in X-ray chest radiographs from the database provided by Japanese Society of Radiological Technology [].

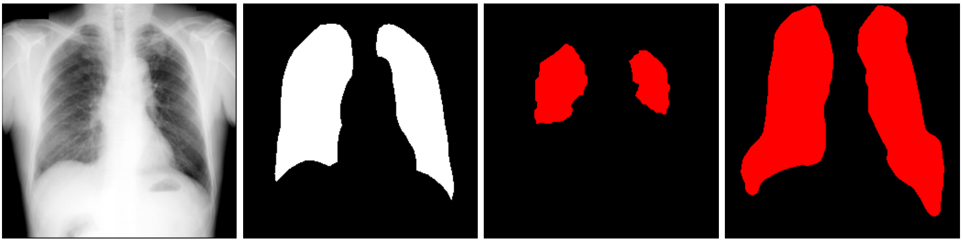

Figure 6: From left to right: Chest radio-graph, ground truth segmentation of lung, GrabCut segmentation, smooth boundary + atlas model segmentation.

The ”local“ latent variables where chosen to have a co-domain K_2 with 18 possible values and receptive fields of size 7×7 pixels in one case and 9×9 pixels in the other one. We also considered different component numbers (4 and 12) for the global latent variable. The models were learned on 124 randomly chosen ground truth segmentations from the database. We used the ”naive Bayes“ initialisation to speed up the learning. The appearance model was chosen to be conditionally pixel-wise independent given the segmentation. The grey-value distributions for the two segment labels are assumed as mixtures of (three) Gaussians each and were learned semi-supervised for each test image (the remaining 123 images from the database) separately. For this, the segmentation was fixed in regions for which the learned atlas mixture predicts a unique a-priory decision. Slightly bigger regions (80% sure decision of the atlas mixture) were used for learning the initial appearance model.

We have used the standard GrabCut method [9] as baseline. Notice, that the underlying model is an MRF on a lattice without latent variables. The parameters of the appearance model were learned semi-supervised by fixing the segmentation in the same ”unique decision“ regions. Table 1 shows the average segmentation precision and its variance obtained by the models with different receptive fields and different number of labels for the global latent variable.

	GC	7x7/4	7x7/12	9x9/4	9x9/12
mean	0.521	0.822	0.836	0.829	0.839
var.	0.117	0.072	0.068	0.073	0.067

Table 1: Lung segmentation precision (dice metric)

It is clearly seen that the considered model class outperforms GrabCut substantially. Not surprisingly, the results are the better the bigger the receptive fields of the local latent variables (responsible for smooth boundaries) and the larger the co-domain of the global latent variable (responsible for the anatomical atlas).

5 Conclusions

The main contribution of the paper was to introduce an alternative method for unsupervised (parameter) learning of general MRFs on bipartite graphs. The modified EM algorithm converges an order of magnitude faster than standard stochastic gradient methods. This opens perspectives to tackle important problems like *e.g.* unsupervised structure learning of

MRFs. It remains, however, to prove theoretically that the modified EM algorithm enjoys similar properties as the standard EM algorithm.

6 Acknowledgements

The authors were supported by the Grant Agency of the Czech Republic under Project P202/12/2071.

References

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

[2] Andrei A. Bulatov and Martin Grohe. The complexity of partition functions. *Theor. Comput. Sci.*, 348(2-3):148–186, 2005.

[3] Uri Heinemann and Amir Globerson. What cannot be learned with Bethe approximations. In Fabio Gagliardi Cozman and Avi Pfeffer, editors, *UAI*, pages 319–326. AUAI Press, 2011.

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

[5] Jeremy Jancsary, Sebastian Nowozin, Toby Sharp, and Carsten Rother. Regression tree fields - an efficient, non-parametric approach to image labeling problems. In *CVPR*, pages 2376–2383. IEEE, 2012.

[6] Pushmeet Kohli, M. Pawan Kumar, and Philip H. S. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[7] 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.

[8] Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005.

[9] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

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

[11] Junji Shiraishi, Shigehiko Katsuragawa, Junpei Ikezoe, Tsuneo Matsumoto, Takeshi Kobayashi, Ken ichi Komatsu, Mitate Matsui, Hiroshi Fujita, Yoshie Kodera, and Kunio Doi. Development of a digital image database for chest radiographs with and without a lung nodule. *American Journal of Roentgenology*, 174(1):71–74, 2000.

[12] Alan D. Sokal. Monte carlo methods in statistical mechanics: Foundations and new algorithms. Lectures notes, 1989.

[13] 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.

[14] T. Tieleman and G.E. Hinton. Using Fast Weights to Improve Persistent Contrastive Divergence. In *Proceedings of the 26th international conference on Machine learning*, pages 1033–1040. ACM New York, NY, USA, 2009.

[15] 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.