Neural Learning for Distributions on Categorical Data

F.X. Albizuri, A.I. Gonzalez, M. Graña, A. d'Anjou

University of the Basque Country

Informatika Fakultatea, P.K. 649, 20080 Donostia, Spain

E-mail: ccpalirx@si.ehu.es; Fax: + 34 943 219306

Abstract. In this paper we define a Boltzmann machine for modelling probability distributions on categorical data, that is, distributions on a set of variables with a finite discrete range. The distribution model is suggested by the log-linear models and it is a generalization of the binary Boltzmann machine. High-order connections are defined instead of hidden units in order to model general probability distributions on multi-valued units. We deduce the iterative learning rule that minimizes the divergence function, which corresponds to a neural scheme. We show that this learning rule converges to the global minimum of the Kullback-Leibler divergence. An example is provided to illustrate the modelling capability of the Boltzmann machine with discrete (non-binary) units.

Keywords: Boltzmann machines, categorical data, discrete units, log-linear models, neural networks

1 Introduction

The Boltzmann machine derives from the Hopfield memory with stochastic dynamics [18, 8]; Boltzmann machines are recurrent neural networks and their dynamics is similar to the stochastic associative memory dynamics. However, Boltzmann machines [16, 2, 17, 1] are not defined to store memories, but to model probability distributions on binary units through a neural learning algorithm. An important characteristic of these neural networks is the stochastic simulation [22] in the learning phase and in the subsequent probabilistic inference. The simulation method is a variant of the Metropolis algorithm [21], a basic Monte Carlo method with a Markov chain, this variant being equivalent to the Gibbs sampler [13]. The stochastic simulation becomes crucial when the network size grows, so this neural network is a massively parallel computation algorithm for probabilistic modelling.

In general, the Boltzmann machine is defined as a probabilistic classifier: given the input we have a probability distribution on the class binary units, a distribution that is typically obtained through a stochastic simulation of the corresponding Markov chain. Alternatively, if input units are suppressed the Boltzmann machine can be viewed as a model for probability distributions on a set of binary units. This approach to the Boltzmann machine is closely related to the log-linear or exponential models [12, 19], developed in the statistical theory to describe especially categorical data [3, 9, 24]. In the statistical theory of log-linear models the parameters of the model are basically determined with

the Newton-Raphson procedure, which requires matrix inversion. Another procedure is the Deming-Stephan algorithm or the iterative proportional scaling algorithm, which does not determine the model parameters but successively adjusts the marginal counts of the contingency table. These algorithms are clearly computationally impracticable when the number of variables (units) is great, and furthermore the Boltzmann machine learning algorithm allows a heuristic schedule that accelerates the convergence. Therefore the advantage of connectionist paradigms such as the Boltzmann machine is the distributed computation of the model parameters. In this context the Boltzmann machine determines the connection weights through steepest descent, which results in its known neural learning rule.

The aim of this paper is to develop a Boltzmann machine for modelling distributions on categorical data, so the binary variables are generalized by defining variables with a finite discrete range. We present a log-linear model suitable for the Boltzmann machine, and we deduce the corresponding neural learning rule as well as its convergence properties. Higher-order Boltzmann machines with binary units have been reported in the literature [23, 7], but they were not generalized in order to consider distributions on discrete data. On the other hand, in [11] and [20] the extension to multi-valued variables is considered; however, the model presented in the former reference is limited to two-order interactions, and the latter develops a physical model where the objective is not to describe a probability distribution. Consequently, these previous works on Boltzmann

machines do not provide a general model for distributions on categorical data. In this paper we propound a Boltzmann machine generalization that can model any distribution on discrete data preserving the neural learning algorithm.

If the Boltzmann machine is limited to two-order interactions we need hidden units to model a general probability distribution. Conversely, hidden units are not necessary for distribution modelling when higher-order connections are defined. It is well known that hidden units introduce local minima into the function minimized by the learning rule, the Kullback-Leibler divergence between the distribution to learn and the model distribution, so that the learning algorithm can obtain non-optimal parameter weights. When high-order connections are used instead of hidden units there is not any local minima [4]. On the other hand, the connection structure of a Boltzmann machine without hidden units can be derived from qualitative properties of the probability distribution [6]. Consequently, in this paper we will define and study a higher-order Boltzmann machine with multi-valued units and without hidden units. The model can be extended to include hidden units, but this extension would not conserve all the properties of the model without hidden units.

We complete the theoretical study with a significant example where a distribution on discrete units is modelled, showing the capability of the Boltzmann machine for learning a probability distribution on categorical data.

In Section 2 we describe the high-order Boltzmann machine with discrete units, showing that it can model any positive distribution on multi-valued units.

The learning rule is derived in Section 3, where convergence properties of this rule are analysed. We provide the learning example in Section 4, and finally, we signal the conclusions in Section 5.

2 The Boltzmann machine with discrete units

The original Boltzmann machine was defined with hidden units and two-order connections, however high-order connections can be used instead of hidden units obtaining interesting convergence properties for the learning rule [4]. In our theory hidden units are not used, but the extension of the distribution model to this case is straightforward.

In the usual Boltzmann machine the probability function is the normalized exponential of a (minus) energy that is the summation of connection weights between activated binary units. For discrete units the naive approach is to multiply connection weights by unit values, but the resultant model is not valid for general distributions [15]. Lin et Lee [20] propose a multi-valued Boltzmann machine based on the spin glass model, but the scheme is not suitable to describe distributions. Anderson and Titterington [11] define a two-order Boltzmann machine with multi-valued units, so that hidden units are needed to model probability distributions; therefore local minima are present and the convergence properties of our model do not hold in this extension of the Boltzmann machine.

Our Boltzmann machine with discrete units is suggested by the log-linear

models, where the logarithm of the probability function is a general function that is linear in the parameters. Now the interactions between units are not connection weights but functions on finite discrete spaces that are defined by a set of parameters. In order to determine the parameters of a log-linear model it is necessary to establish some constraints on the parametrization so that the parameters are independent and the divergence function can be minimized with the iterative method of steepest descent (or Newton-Raphson). These constraints can be established in several different ways and we will describe a model that is a generalization of the binary Boltzmann machine.

The probability function of our Boltzmann machine is

$$P(\mathbf{x}) = \frac{1}{Z} \exp \sum_{\alpha \in A^*} u_\alpha(\mathbf{x}) \tag{1}$$

where

$$Z = \sum_{\mathrm{x}} \exp \sum_{\alpha \in A^*} u_\alpha(\mathbf{x})$$

The configuration of the units is $\mathbf{x} = (x_1, \ldots, x_d)$, where each x_i has a finite discrete range. In these formulas A^* is the set of connections α, and each connection α is a nonempty subset of $\{1, \ldots, d\}$. In a two-order network a connection joins at most two units, and on the other hand, a higher-order network has some connections between three or more units.

The interaction associated with a connection α is given by a function $u_\alpha(\mathbf{x})$, which depends on $\mathbf{x}$ through the components of $\mathbf{x}$ corresponding to α, that is $u_\alpha(\mathbf{x}) = u_\alpha(\mathbf{x}_\alpha)$. So each function $u_\alpha(\mathbf{x}_\alpha)$ is defined on a finite discrete space

of partial configurations and it is determined by a finite number of parameters. Besides, $u_\alpha(\mathbf{x}) = 0$ if $x_i = 0$ for any $i \in \alpha$, where $x_i = 0$ is a reference value of the variable. It is also required that given a connection $\alpha \in A^*$ of the model, any subset $\beta \subset \alpha$ of this connection is also a connection of the model, $\beta \in A^*$.

The logarithm of the probability function can be written as

$$\log P(\mathbf{x}) = \sum_{\alpha \in A} u_\alpha(\mathbf{x}) \tag{2}$$

where $A = A^* \cup \{\emptyset\}$ and $u_\emptyset = -\log Z$.

Any positive distribution $P(\mathbf{x})$ on a finite discrete space has a log-linear expression (2), where the connections are in general all the subsets of $\{1, \ldots, d\}$. It is easy to show that if we take all possible configurations $\mathbf{x}$ in (2) it leads to a triangular equation system, that can be solved by backsubstitution, so that given a probability distribution the model parameters are determined. In practice, in order to model an empirical distribution we will consider in (1) interactions up to an order r smaller than d when the number of units is large; our objective will not be to reproduce exactly the empirical distribution, but to obtain an approximation distribution that minimizes some function of the model parameters, specifically the Kullback-Leibler divergence.

We give an example to illustrate the notation introduced. The two-order probability function (1) for a distribution on $\mathbf{x} = (x_1, x_2, x_3)$, $x_i \in \{0, 1, 2\}$, is the normalized exponential of

$$u_1(x_1) + u_2(x_2) + u_3(x_3) + u_{12}(x_1, x_2) + u_{13}(x_1, x_3) + u_{23}(x_2, x_3)$$

The parameters of the model are $u_1(1)$, $u_1(2)$, ..., $u_{12}(1,1)$, $u_{12}(1,2)$, $u_{12}(2,1)$, $u_{12}(2,2)$, etc.

Finally, we note that for binary units, $x_i \in \{0,1\}$, the model (1) is just the usual Boltzmann machine probability function.

3 Learning and convergence properties

Now, we will deduce the neural learning rule for this Boltzmann machine. The objective is to minimize the Kullback-Leibler divergence between a positive distribution $P^s(\mathbf{x})$ that we want to learn, usually given by a set of samples, and the model distribution $P(\mathbf{x})$,

$$D = \sum_{\mathrm{x}} P^s(\mathbf{x}) \log \frac{P^s(\mathbf{x})}{P(\mathbf{x})}$$

The divergence D is a function of the parameters $u_\alpha(\mathbf{y}_\alpha)$. (We will use $\mathbf{y}$ in the argument of the model parameters and we will reserve $\mathbf{x}$ for the summation index.) Partial derivatives are calculated as in the binary case, obtaining

$$\frac{\partial D}{\partial [u_\alpha(\mathbf{y}_\alpha)]} = \tau_\alpha(\mathbf{y}_\alpha) - \tau^s_\alpha(\mathbf{y}_\alpha)$$

where

$$\tau_\alpha(\mathbf{y}_\alpha) = \sum_{\mathrm{x}\,:\,\mathrm{x}_\alpha = \mathrm{y}_\alpha} P(\mathbf{x})$$

and

$$\tau^s_\alpha(\mathbf{y}_\alpha) = \sum_{\mathrm{x}\,:\,\mathrm{x}_\alpha = \mathrm{y}_\alpha} P^s(\mathbf{x})$$

The iterative learning rule is the steepest descent:

$$[u_\alpha(\mathbf{y}_\alpha)]_{t+1} = [u_\alpha(\mathbf{y}_\alpha)]_t - \rho[\tau_\alpha(\mathbf{y}_\alpha) - \tau^s_\alpha(\mathbf{y}_\alpha)]$$

for all $\alpha \in A^*$ and for all $\mathbf{y}_\alpha$ without null components.

This learning rule is neural because, in order to obtain the $\tau_\alpha(\mathbf{y}_\alpha)$ at each step through a stochastic simulation of the Markov chain corresponding to the distribution (1), we can compute simultaneously at each connection $\alpha \in A^*$ the frequency of the configurations $\mathbf{x}$ such that $\mathbf{x}_\alpha = \mathbf{y}_\alpha$ for every $\mathbf{y}_\alpha$.

Convergence properties of our Boltzmann machine for distributions on discrete data can be obtained by extending the convergence properties of the binary high-order Boltzmann machine without hidden units proved by Albizuri et al. [4]. We will not reproduce the proofs but we will give an outline of this theory and will enunciate the main results. Rigorous proofs can be constructed by following the proofs of this reference and developing a parallel reasoning.

The second-order derivative of the divergence is given by

$$\frac{\partial^2 D}{\partial[u_\alpha(\mathbf{y}_\alpha)]\partial[u_\beta(\mathbf{z}_\beta)]} = \tau_{\alpha,\beta}(\mathbf{y}_\alpha, \mathbf{z}_\beta) - \tau_\alpha(\mathbf{y}_\alpha)\tau_\beta(\mathbf{z}_\beta)$$

where

$$\tau_{\alpha,\beta}(\mathbf{y}_\alpha, \mathbf{z}_\beta) = \sum_{\mathrm{x}\,:\,\mathrm{x}_\alpha = \mathrm{y}_\alpha \wedge \mathrm{x}_\beta = \mathrm{z}_\beta} P(\mathbf{x})$$

If we define the random variable $X(\mathbf{y}_\alpha)$ such that it takes value 1 when $\mathbf{x}_\alpha = \mathbf{y}_\alpha$ and 0 otherwise, then

$$\frac{\partial^2 D}{\partial[u_\alpha(\mathbf{y}_\alpha)]\partial[u_\beta(\mathbf{z}_\beta)]} = \mathrm{Cov}(X(\mathbf{y}_\alpha), X(\mathbf{z}_\beta))$$

where the covariance is under $P(\mathbf{x})$, and it can be shown that the divergence D, a function of the model parameters, has a positive definite Hessian matrix and consequently it is a convex function.

Since the second-order derivative is bounded, $|\tau_{\alpha,\beta}(\mathbf{y}_\alpha, \mathbf{z}_\beta) - \tau_\alpha(\mathbf{y}_\alpha)\tau_\beta(\mathbf{z}_\beta)| \leq 1$, it can be proved (from the Taylor's theorem) that if $\rho < 2/|L|^2$, $|L|$ being the number of parameters, then the divergence decreases at each step of the learning algorithm, and the parameter values generated being bounded, it can be established the convergence of the learning rule to the global minimum of the divergence.

This value of ρ is theoretical, and it can be deduced from the convexity of the divergence that whenever $\tau_\alpha(\mathbf{y}_\alpha)$ converges to $\tau^s_\alpha(\mathbf{y}_\alpha)$ for any α and $\mathbf{y}_\alpha$ (without null components) during the learning process, then the parameter values converge to the global minimum of the divergence. In practice we rely on this last result to set the learning rule, and ρ is fixed experimentally in order that $||\nabla D||^2 = \sum_{\alpha,\mathrm{y}_\alpha} (\tau_\alpha(\mathbf{y}_\alpha) - \tau^s_\alpha(\mathbf{y}_\alpha))^2$ converges to 0.

Consequently, the high-order Boltzmann machine for multi-valued units, without hidden units, defines a convex divergence, and the neural learning algorithm converges to the global minimum. These remarkable properties do not hold when hidden units are used in a two-order model.

4 An example

We present an example where the Boltzmann machine is used to learn a distribution on non-binary units. Our example is similar to the problem studied in [17], where a Boltzmann machine with binary units and hidden units was used. In [5] this problem was treated with a high-order binary Boltzmann machine. Now we will define a problem where non-binary units are required. In these examples the basic problem is to implement logical relations [14] between discrete variables through a probability distribution. The neural network determines the model parameters by learning from this distribution and subsequently we test the neural network to check whether the logical relations have been modelled.

We will consider an array of four units, each unit taking the values *a(= 0), b, c, d, e, f*. We say that an array configuration is well ordered if the letters in the array are consecutive (a follows f), as in *bcde, efab*, etc. The distribution to learn is defined so that well ordered configurations have higher probability than the configurations that are not well ordered. Then the Boltzmann machine parameters are determined by means of its learning rule.

In order to test the distribution learned by the Boltzmann machine we will fix the value of a unit and compute the probability of the configuration for the rest of units that completes a well ordered array configuration. (This probability coincides with the conditional probability corresponding to the value of the specified unit.) In the original distribution, the well ordered configuration has a higher conditional probability than the conditional probabilities of the

rest of configurations; if this property holds in the learned distribution we will conclude that the Boltzmann machine has modelled satisfactorily the original distribution.

In this experiment, the distribution to learn assigns to the well ordered configurations a probability mass that adds up to 0.95. For the approximation distribution model of our Boltzmann machine we have defined two-order interactions between adjacent units (and one-order interactions in each unit). We have determined the parameters through 500 iterations of the learning rule, with $\rho = 1$.

In Table 1 we report the test results: the row indicates the unit fixed, the column the value assigned, and the corresponding entry is the probability of the partial configuration that completes a well ordered array. The approximation distribution of the Boltzmann machine clearly models the distribution to learn. We point out that with the first 160 iterations the probabilities obtained are greater than 0.5, the learning process is very effective.

5 Conclusion

In this paper we have defined a Boltzmann machine for modelling probability distributions on (finite range) discrete data. The distribution model is log-linear and the required parametrization constraints are defined so that it can be viewed as a generalization of the binary Boltzmann machine. Any distribution can be

Table 1: Probability of the partial configuration that completes a well ordered array when the value of a unit is fixed.

Value	a	b	c	d	e	f
Unit 1	0.808	0.867	0.867	0.821	0.760	0.765
Unit 2	0.744	0.827	0.866	0.866	0.819	0.766
Unit 3	0.744	0.766	0.819	0.866	0.866	0.827
Unit 4	0.808	0.765	0.760	0.821	0.867	0.867

expressed with this model, in general taking the full set of connections. With interactions up to a determined order the Boltzmann machine provides an approximation distribution of the distribution to learn. We have also deduced the neural learning rule, which converges to the global minimum of the divergence. Convergence properties of the Boltzmann machine for distributions on discrete data can be obtained from the convergence properties of the binary high-order Boltzmann machine without hidden units.

We remark that the convergence to the global minimum does not hold when hidden units are introduced into the two-order Boltzmann machine with binary or multi-valued units to model general probability distributions. On the other hand, the Newton-Raphson algorithm and similar two-order methods do not have the convergence properties of our Boltzmann machine.

Acknowledgments

This work was supported by the research grants PI 1998/21 and UE 1999/1 from the Basque Government and MAT 99-1049-C03-03 from the CICYT. The authors are grateful to the reviewers for their comments.

References

[1] E. H. L. Aarts and J. H. M. Korst. *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, New York, 1989.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[3] A. Agresti. *Categorical Data Analysis*. Wiley, New York, 1990.

[4] F. X. Albizuri, A. d'Anjou, M. Grana, and J. A. Lozano. Convergence properties of High-Order Boltzmann Machines. *Neural Networks*, 9(9):1561–1567, 1996.

[5] F. X. Albizuri, A. d'Anjou, M. Grana, F. J. Torrealdea, and M. C. Hernandez. The high-order Boltzmann Machine: Learned distribution and topology. *IEEE Transactions on Neural Networks*, 6(3):767–770, 1995.

[6] F.X. Albizuri, A. d'Anjou, M. Grana, and P. Larranaga. Structure of the high-order Boltzmann machine from independence maps. *IEEE Transactions on Neural Networks*, 8(6):1351–1358, 1997.

[7] S. Amari, K. Kurata, and H. Nagaoka. Information geometry of Boltzmann machines. *IEEE Transactions on Neural Networks*, 3(2):260–271, 1992.

[8] D. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55:1530–1533, 1985.

[9] E.B. Andersen. *The Statistical Analysis of Categorical Data*. Springer-Verlag, Berlin, second edition, 1991.

[10] J. A. Anderson and E. Rosenfeld, editors. *Neurocomputing: Foundations of Research*. MIT Press, Cambridge, MA, 1988.

[11] N.H. Anderson and D.M. Titterington. Beyond the binary Boltzmann machine. *IEEE Transactions on Neural Networks*, 6(5):1229–1236, 1995.

[12] J.N. Darroch, S.L. Lauritzen, and T.P. Speed. Markov fields and log linear interaction models for contingency tables. *Annals of Statistics*, 8:522–539, 1980.

[13] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[14] M. Grana, A. d'Anjou, F.X. Albizuri, M. Hernandez, F.J. Torrealdea, A. de la Hera, and A.I. Gonzalez. Experiments of fast learning with high order Boltzmann machines. *Applied Intelligence*, 7:287–303, 1997.

[15] M. Grana, V. Lavin, A. d'Anjou, F.X. Albizuri, and J.A. Lozano. High-order Boltzmann machines applied to the Monk's problems. In *European Symposium on Artificial Neural Networks*, pages 117–122, Brussels, Belgium, 1994.

[16] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 448–453, 1983.

[17] G. E. Hinton and T. J. Sejnowski. Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 282–317. MIT Press, Cambridge, MA, 1986.

[18] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79:2554–2558, 1982. Reprinted in Anderson and Rosenfeld [10].

[19] S. L. Lauritzen. Lectures on contingency tables. Technical Report R 89-24, The University of Aalborg, Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg, Denmark, 1989.

[20] C.T. Lin and C.S.G. Lee. A multi-valued Boltzmann machine. *IEEE Transactions on Systems, Man and Cybernetics*, 25(4):660–669, 1995.

[21] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations for fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[22] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[23] T. J. Sejnowski. Higher-order Boltzmann machines. In J.S. Denker, editor, *AIP Coference Proceedings 151, Neural Networks for Computing*, pages 398–403, Snowbird, UT, 1986.

[24] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, New York, 1990.

BIOGRAPHIES

F. Xabier Albizuri received the M.Sc. degree in physics in 1987 and the Ph.D. degree in computer science in 1995 from the University of the Basque Country, Spain. He is a Professor of the Department of Computes Science and Artificial Intelligence. His current research interest is on neural networks, pattern recognition, graphical models.

Ana Isabel Gonzalez received the M.Sc. degree in computer science in 1993 from the University of the Basque Country. She is a Ph.D. student. Her research includes competitive neural networks and image processing.

Manuel Graña obtained the M.Sc. and Ph.D. degrees in computer science from the University of the Basque Country in 1982 and 1989 respectively. He is a Professor of the Department of Computer Science and Artificial Intelligence. His current interests include neural networks for digital signal processing.

Alicia d'Anjou received the M.Sc. and the Ph.D. degrees in physics from the University of Navarra in 1974 and 1978 respectively. She is a Professor of the Department of Computer Science and Artificial Intelligence. Her current interests include adaptive algorithms and neural networks.