# Prediction of breast cancer recurrence using Classification Restricted Boltzmann Machine with *Dropping*

**Jakub M. Tomczak**
Wrocław University of Technology
Wrocław, Poland
jakub.tomczak@pwr.wroc.pl

## Abstract

In this paper, we apply Classification Restricted Boltzmann Machine (ClassRBM) to the problem of predicting breast cancer recurrence. According to the Polish National Cancer Registry, in 2010 only, the breast cancer caused almost $25\%$ of all diagnosed cases of cancer in Poland. We propose how to use ClassRBM for predicting breast cancer return and discovering relevant inputs (symptoms) in illness reappearance. Next, we outline a general probabilistic framework for learning Boltzmann machines with masks, which we refer to as *Dropping*. The fashion of generating masks leads to different learning methods, i.e., *DropOut*, *DropConnect*. We propose a new method called *DropPart* which is a generalization of *DropConnect*. In *DropPart* the Beta distribution instead of Bernoulli distribution in *DropConnect* is used. At the end, we carry out an experiment using real-life dataset consisting of 949 cases, provided by the Institute of Oncology Ljubljana.

## 1 Introduction

Machine learning algorithms has been successfully applied to many complex problems, especially in medical domain [5], e.g., in diabetes treatment [12], in predicting breast cancer recurrence [11]. In this paper, we would like to focus on the second problem since according to Polish National Cancer Registry, in 2010 only, $25\%$ of all cases of cancer were associated with the breast cancer [1]. Current diagnostics techniques are unable to successfully predict breast cancer return, therefore, there is a constant need to develop new predictive models. According to the author's knowledge, Classification Restricted Boltzmann Machine (ClassRBM) has been not yet applied to the prediction of the breast cancer recurrence. Moreover, the ClassRBM can be used to determine relevant symptoms of the illness reappearance. These two aspects constitute an original contribution to the field of machine learning and modeling of biomedical phenomena.

In this paper, only a preliminary study on learning ClassRBM with *Dropping* is presented. However, the paper aims at making the following contribution:

- We propose a general probabilistic framework for learning ClassRBM with masks, which we call *Dropping*. The fashion of generating masks leads to different learning methods, namely, *DropOut* [4, 10] and *DropConnect* [13].
- We propose a new learning method which is a generalization of *DropConnect*. The connections are partially removed during learning, i.e., an activation of each connection is drawn with Beta distribution (with $a, b \leq 1$). We call this method *DropPart*.
- We use ClassRBM to discover relevant inputs, or symptoms in the medical domain.

- We carry out an experiment using real-life dataset consisting of 949 cases, provided by the Institute of Oncology Ljubljana [11].

The paper is structured as follows. In Section 2.1 Classification Restricted Boltzmann Machine is outlined. In Section 2.2 the conditional distribution for prediction is given. In Section 2.2 the conditional distribution for discovering relevant inputs is described. In Section 2.4 learning with *Dropping* is outlined. At the end of the paper, an experiment that examines the predictive and explanatory capabilities of ClassRBM in in the problem of breast cancer recurrence is carried out. Obtained results within the experiment are discussed and conclusions are drawn.

## 2 Classification Restricted Boltzmann Machine

### 2.1 The model

Restricted Boltzmann Machine (RBM) is a two-layer undirected graphical model where the first layer consists of visible input variables $\mathbf{x} \in \{0,1\}^D$, and the second layer consists of hidden variables (units) $\mathbf{h} \in \{0,1\}^M$. We allow only the inter-layer connections, i.e., there are no connections within layers. Moreover, we add a third layer that represents observable output variable $y \in \{1, 2, \ldots, K\}$. Further, we use the 1-to-$K$ coding scheme which results in representing output as a binary vector of length $K$ denoted by $\mathbf{y}$, such that if the output (or class) is $k$, then all elements are zero except element $y_k$ which takes the value 1.

A RBM with $M$ hidden units is a parametric model of the joint distribution of visible and hidden variables, that takes the form:

$$p(\mathbf{x}, \mathbf{y}, \mathbf{h}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} e^{-E(\mathbf{x},\mathbf{y},\mathbf{h}|\boldsymbol{\theta})} \tag{1}$$

with parameters $\boldsymbol{\theta} = \{\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{W}^1, \mathbf{W}^2\}$, and where:

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}|\boldsymbol{\theta}) = -\mathbf{b}^\top \mathbf{x} - \mathbf{c}^\top \mathbf{h} - \mathbf{d}^\top \mathbf{y} - \mathbf{x}^\top \mathbf{W}^1 \mathbf{h} - \mathbf{h}^\top \mathbf{W}^2 \mathbf{y} \tag{2}$$

is an energy function, and

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x},\mathbf{y},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{y},\mathbf{h}|\boldsymbol{\theta})} \tag{3}$$

is a partition function.

This model is called Classification Restricted Boltzmann Machine (ClassRBM)[1] and is argued to be used as a stand-alone non-linear classifier [7]. The main advantage of using RBM as a classifier is that it remains all generative advantages of RBM and additionally it allows to calculate distribution $p(y|\mathbf{x})$ straightforwardly. It can be shown that the following expressions hold true for ClassRBM [6, 7]:[2]

$$p(\mathbf{x}|\mathbf{h}) = \prod_i p(x_i|\mathbf{h}) \tag{4}$$

$$p(x_i = 1|\mathbf{h}) = \mathrm{sigm}(b_i + \mathbf{W}^1_{i\cdot}\mathbf{h}) \tag{5}$$

$$p(y|\mathbf{h}) = \frac{e^{d_y + (\mathbf{W}^2_{\cdot y})^\top \mathbf{h}}}{\sum_{\overline{y}} e^{d_{\overline{y}} + (\mathbf{W}^2_{\cdot \overline{y}})^\top \mathbf{h}}} \tag{6}$$

$$p(\mathbf{h}|y, \mathbf{x}) = \prod_j p(h_j|\mathbf{h}) \tag{7}$$

$$p(h_j = 1|y, \mathbf{x}) = \prod_j \mathrm{sigm}(c_j + W^2_{jy} + (\mathbf{W}^1_{\cdot j})^\top \mathbf{x}) \tag{8}$$

[1] The ClassRBM was first proposed in [6].

[2] Further in the paper, sometimes we omit explicit conditioning on parameters $\boldsymbol{\theta}$.

where $\mathrm{sigm}(\cdot)$ is the logistic sigmoid function, $\mathbf{W}^{\ell}_{i\cdot}$ is $i^{\text{th}}$ row of weights matrix $\mathbf{W}^{\ell}$, $\mathbf{W}^{\ell}_{\cdot j}$ is $j^{\text{th}}$ column of weights matrix $\mathbf{W}^{\ell}$, $W^{\ell}_{ij}$ is the element of weights matrix $\mathbf{W}^{\ell}$.

### 2.2 Prediction

It is possible to exactly compute the distribution $p(y|\mathbf{x})$ which can be further used to choose the most probable class label. This conditional distribution takes the following form [6, 7]:

$$p(y|\mathbf{x}) = \frac{e^{d_y} \prod_j \left(1 + \left(e^{c_j + (\mathbf{W}^1_{\cdot j})^\top \mathbf{x}}\right) e^{W^2_{jy}}\right)}{\sum_{\overline{y}} e^{d_{\overline{y}}} \prod_j \left(1 + \left(e^{c_j + (\mathbf{W}^1_{\cdot j})^\top \mathbf{x}}\right) e^{W^2_{j\overline{y}}}\right)}. \tag{9}$$

Pre-computing the terms $c_j + (\mathbf{W}^1_{\cdot j})^\top \mathbf{x}$ allows to reduce the time needed for computing the conditional distribution to $O(MD + MK)$ [6, 7].

### 2.3 Discovering relevant inputs

We can also use the ClassRBM to determine relevancy of inputs by calculating conditional probabilities $p(x_i = 1|\mathbf{x}_{\setminus i}, y)$ [7]. In the medical context this conditional probability expresses the probability of occurring $i^{\text{th}}$ input for given other inputs (symptoms) and class label. For example, if $y = 1$ denotes breast cancer recurrence, we can quantitatively determine which inputs are important (relevant) during prediction. Further, we make an assumption that for $i^{\text{th}}$ input all other inputs are inactive, i.e, $\mathbf{x}_{\setminus i} = \mathbf{0}$. Hence, we get the following expression for the conditional probability:

$$\begin{aligned} p(x_i = 1|\mathbf{x}_{\setminus i} = \mathbf{0}, y) &= \frac{e^{b_i} \prod_j \left(1 + e^{c_i + (\mathbf{W}^1_{\cdot i})^\top \mathbf{x} + W^2_{jy}}\right)}{\sum_{x_i = \{0,1\}} e^{b_i} \prod_j \left(1 + e^{c_j + (\mathbf{W}^1_{\cdot j})^\top \mathbf{x} + W^2_{j\overline{y}}}\right)} \\ &= \frac{e^{b_i} \prod_j \left(1 + e^{c_j + (W^1_{ij}) + W^2_{jy}}\right)}{e^{b_d} \prod_j \left(1 + e^{c_j + (W^1_{ij}) + W^2_{jy}}\right) + \prod_j \left(1 + e^{c_j + W^2_{jy}}\right)} \end{aligned} \tag{10}$$

Calculating the conditional probabilities for all inputs using Equation 10 leads to discovering relevant inputs. For example, we may choose a threshold, e.g., equal $0.5$, and inputs with probabilities which are larger than the threshold are supposed to be important in the prediction.

### 2.4 Learning with *Dropping*

Typically, the parameters $\boldsymbol{\theta}$ in ClassRBM are learned from data using the likelihood function:

$$p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{h}} e^{-E(\mathbf{x}, \mathbf{y}, \mathbf{h}|\boldsymbol{\theta})}. \tag{11}$$

There are different inductive principles for learning ClassRBM [8], e.g., Approximate Maximum Likelihood, Maximum Pseudo-Likelihood, Ratio Matching. However, the widely-used method is Contrastive Divergence [2, 3], which is further used in this paper.

Recently, new learning methods were introduced which perform a kind of regularization during learning RBM, namely, *DropOut* [4, 10] and *DropConnect* [13]. In this paper, we would like to address this kind of learning and give a general approach called *Dropping* which explains theoretical foundations of these two methods and gives rise to a new method which we refer to as *DropPart*.

Let us introduce a *mask*, $\mathcal{M} = \{\mathbf{M}^1, \mathbf{M}^2, \mathbf{m}\}$, that determines which connections should be active during one learning iteration ($\mathbf{M}^1$ – determines connections in weights $\mathbf{W}^1$, $\mathbf{M}^2$ – determines connections in weights $\mathbf{W}^1$, and $\mathbf{m}$ – connections in bias $\mathbf{c}$). Then, we get (symbol $\star$ denotes Hadamard product, called also element-wise product):

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) &= \sum_{\mathcal{M}} p(\mathbf{x}, \mathbf{y}, \mathcal{M}|\boldsymbol{\theta}) \\ &= \sum_{\mathcal{M}} p(\mathbf{x}, \mathbf{y}|\mathbf{b}, \mathbf{m} \star \mathbf{c}, \mathbf{d}, \mathbf{M}^1 \star \mathbf{W}^1, \mathbf{M}^2 \star \mathbf{W}^2)\, p(\mathcal{M}|\boldsymbol{\theta}). \end{aligned} \tag{12}$$

We may assume unconditional independence $\mathcal{M} \perp \boldsymbol{\theta} \mid \emptyset$ which yields

$$p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = \sum_{\mathcal{M}} p(\mathbf{x}, \mathbf{y}|\mathbf{b}, \mathbf{m} \star \mathbf{c}, \mathbf{d}, \mathbf{M}^1 \star \mathbf{W}^1, \mathbf{M}^2 \star \mathbf{W}^2)\, p(\mathcal{M}). \tag{13}$$

In general, we refer the approach of including mask in further inference to as *Dropping*. The way the masks are generated yields different schemas:

- In *Dropout* choosing a unit to be active is a Bernoulli random variable with probability $p$. Hence, if $i^{\text{th}}$ unit is active, then the $i^{\text{th}}$ row in $\mathbf{M}^\ell$ consists of ones ($\mathbf{M}^\ell_{i\cdot} = \mathbf{1}$), and zeros – otherwise, for $\ell = 1, 2$. Similarly, if $i^{\text{th}}$ is drawn to be active, we set $\mathbf{m}_i = 1$. Typically, $p$ is set to $0.5$ [4].
- In *Dropconnect* choosing a connection to be active is a Bernoulli random variable, i.e., $\mathbf{M}^\ell_{ij} \sim \text{Bernoulli}(p)$, for $\ell = 1, 2$, and $\mathbf{m}_i \sim \text{Bernoulli}(p)$. The empirical results show that for $p = 0.5$ the best performance was obtained [13].
- We propose a new method called *DropPart*. In *DropPart* choosing a connection to be active is a Beta random variable, i.e., $\mathbf{M}^\ell_{ij} \sim \text{Beta}(a, b)$, for $\ell = 1, 2$, and $\mathbf{m}_i \sim \text{Beta}(a, b)$. The motivation for applying Beta distribution is twofold. First, for $a, b \leq 1$ one obtains a smooth version of *DropConnect*. Second, such approach models spreading of bioelectric current in real neurons, i.e., neurons are active only partially (stronger or weaker).

During learning using Contrastive Divergence (or other gradient-based learning rule), it is important to calculate gradient of the log-likelihood function. Let us assume the likelihood function in form as in Equation 13. Then the logarithm of the likelihood function is intractable analytically, however, we can calculate its lower bound using the Jensen's inequality:

$$\begin{aligned}\log p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) &= \log \sum_{\mathcal{M}} p(\mathbf{x}, \mathbf{y}|\mathbf{b}, \mathbf{m} \star \mathbf{c}, \mathbf{d}, \mathbf{M}^1 \star \mathbf{W}^1, \mathbf{M}^2 \star \mathbf{W}^2)\, p(\mathcal{M}) \\ &\geq \sum_{\mathcal{M}} \big( \log p(\mathbf{x}, \mathbf{y}|\mathbf{b}, \mathbf{m} \star \mathbf{c}, \mathbf{d}, \mathbf{M}^1 \star \mathbf{W}^1, \mathbf{M}^2 \star \mathbf{W}^2) + \log p(\mathcal{M}) \big). \end{aligned} \tag{14}$$

Notice that calculating the derivative of the lower bound of the log-likelihood in Equation 14 wrt. $\mathbf{W}^1$, $\mathbf{W}^2$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ is similar to calculating the derivative of the standard likelihood function in Equation 11 (because of the assumption of unconditional independence $\mathcal{M} \perp \boldsymbol{\theta} \mid \emptyset$ the term $\log p(\mathcal{M})$ disappears) but with the summation over all possible masks $\mathcal{M}$. Moreover, the learning rules remain the same as in standard Contrastive Divergence (or other gradient-based learning procedures) but the mask is included, i.e., $\mathbf{m} \star \mathbf{c}$, $\mathbf{M}^1 \star \mathbf{W}^1$, and $\mathbf{M}^2 \star \mathbf{W}^2$.

During one learning iteration, in *DropOut* and *DropConnect* a crude approximation of the lower bound of the log-likelihood is used because the summation is replaced by one randomly drawn mask. Similarly, we use only one randomly drawn mask in *DropPart*.

Since we have applied mask to learning, we should use mask in prediction. However, we omit this issue and leave it for further research. In this paper, we make predictions using the conditional distribution in Equation 9. Only for *DropOut* we apply the distribution in Equation 9 with weights $\mathbf{W}^2$ replaced by $\mathbf{W}^2/2$ as indicated in [4].

## 3 Experiments

### 3.1 Preliminaries

In the experiment we use the real-world medical dataset provided by the Institute of Oncology, Ljubljana [11]. The goal is to predict whether the patient will have a recurrence of a breast cancer within 10 years after surgery. Each patient is described by 15 categorical features (details can be found in [11]).

In the experiment we used the *classification accuracy* ($CA$) as the assessment metric. The $CA$ was applied because the authors of [11] have obtained results for two human doctors ($O1$ and $O2$ in Figure 1). The oncologists were asked to predict the class value for randomly chosen 100 cases and

then the $CA$ value was calculated [11]. The obtained quantities by oncologists are worst than those achieved by machine learning methods (see Figure 1) but this fact does not lead to a conclusion that the classifiers have significantly higher accuracy. However, it can give an insight in the usefulness of the application of machine learning methods in the medical domain.

We used the following classifiers in the experiment:

- ClassRBM with Contrastive Divergence learning (ClassRBM);
- ClassRBM with Contrastive Divergence and DropOut learning (ClassRBM+DropOut);
- ClassRBM with Contrastive Divergence and DropConnect learning (ClassRBM+DropConnect);
- ClassRBM with Contrastive Divergence and DropPart learning (ClassRBM+DropPart);
- Classification and Regression Tree (CART);
- Naive Bayes classifier (Naive Bayes);
- Bagging of $50$ CART (Bagging);
- AdaBoost of CART (AdaBoost);
- LogitBoost of CART (LogitBoost);
- SVM with radial basis function (SVM);
- Random Forest of $50$ CART (Random Forest).

ClassRBM and learning procedures were implemented in Matlab and for all other methods Matlab's implementations were used.

In order to have proper comparison of machine learning methods and human oncologists, we have used the original division of data into training set ($70\%$ of cases) and test set (remaining cases). All methods were learned and evaluated using this data division.

ClassRBM was learned using learning rate equal $0.01$ and $0.1$, momentum rate equal $0.5$, and $100000$ number of iterations (no mini-batch technique was applied). Additionally, we used $\mathrm{Bern}(0.5)$ in *DropOut* and *DropConnect*, and three different sets of parameters in *DropPart* with $\mathrm{Beta}(a, b)$, namely, $(a, b) \in \{(0.1, 0.1), (0.5, 0.5), (1, 1)\}$. All catagorical features were binarized resulting in $55$ binary inputs (see Appendix). The experiment was run $10$ times.

### 3.2 Results and Discussion

The results for ClassRBM are given in Tables 1 and 2. The best results for ClassRBM with different learning methods in comparison to other methods used in the experiment are given in Figure 1.

It can be noticed (see Figure 1) that ClassRBM performs comparably to ensemble classifiers (Bagging, AdaBoost and LogitBoost) and slightly better than Naive Bayes and Random Forest. It is remarkable because the ClassRBM obtains not only high classification accuracy but also provides relevant inputs discovery and assigns probabilities to class labels. These advantages of ClassRBM allows to state that it is high quality classifier basing on its quantitative (i.e. classification accuracy) and qualitative performance (i.e. generative capabilities).

It is worth to notice that *DropOut* performed the best in comparison to *DropConnect* and *DropPart*. Additionally, comparing results for *DropConnect* and *DropPart* with $\mathrm{Beta}(0.1, 0.1)$ (see Tables 1 and 2) we can state that indeed *DropPart* behaves very similar to *DropConnect* as predicted. However, *DropPart* performed better than *DropConnect*. Nevertheless, we believe that application of an inference method which approximates the conditional probability in Equation 9 but with sum over masks will give much better results for *DropConnect* and *DropPart*. An approximated method for inference with *DropConnect* is proposed in [13].

Table 1: Results for ClassRBM with different learning methods and learning rate equal $0.01$. Mean values and standard deviations are given.

| Number of hidden units | ClassRBM | ClassRBM +DropOut $\mathrm{Bern}(0.5)$ | ClassRBM +DropConnect $\mathrm{Bern}(0.5)$ | ClassRBM +DropPart $\mathrm{Beta}(0.1, 0.1)$ | ClassRBM +DropPart $\mathrm{Beta}(0.5, 0.5)$ | ClassRBM +DropPart $\mathrm{Beta}(1, 1)$ |
|---|---|---|---|---|---|---|
| 5 | $0.548 \pm 0.115$ | $0.679 \pm 0.037$ | $0.658 \pm 0.017$ | $0.671 \pm 0.026$ | $0.662 \pm 0.047$ | $0.620 \pm 0$ |
| 10 | $0.685 \pm 0.114$ | $0.718 \pm 0.018$ | $0.666 \pm 0.025$ | $0.715 \pm 0.026$ | $0.719 \pm 0.031$ | $0.723 \pm 0.025$ |
| 15 | $0.735 \pm 0.026$ | $0.738 \pm 0.015$ | $0.678 \pm 0.022$ | $0.702 \pm 0.035$ | $0.706 \pm 0.039$ | $0.720 \pm 0.027$ |
| 20 | $0.714 \pm 0.025$ | $0.721 \pm 0.025$ | $0.677 \pm 0.013$ | $0.692 \pm 0.033$ | $0.717 \pm 0.028$ | $0.714 \pm 0.022$ |

Table 2: Results for ClassRBM with different learning methods and learning rate equal $0.1$. Mean values and standard deviations are given.

| Number of hidden units | ClassRBM | ClassRBM +DropOut $\mathrm{Bern}(0.5)$ | ClassRBM +DropConnect $\mathrm{Bern}(0.5)$ | ClassRBM +DropPart $\mathrm{Beta}(0.1, 0.1)$ | ClassRBM +DropPart $\mathrm{Beta}(0.5, 0.5)$ | ClassRBM +DropPart $\mathrm{Beta}(1, 1)$ |
|---|---|---|---|---|---|---|
| 5 | $0.714 \pm 0.044$ | $0.641 \pm 0.108$ | $0.553 \pm 0.119$ | $0.630 \pm 0.063$ | $0.696 \pm 0.034$ | $0.687 \pm 0.050$ |
| 10 | $0.725 \pm 0.011$ | $0.694 \pm 0.014$ | $0.577 \pm 0.104$ | $0.649 \pm 0.022$ | $0.694 \pm 0.016$ | $0.701 \pm 0.016$ |
| 15 | $0.704 \pm 0.028$ | $0.692 \pm 0.025$ | $0.606 \pm 0.080$ | $0.580 \pm 0.102$ | $0.687 \pm 0.019$ | $0.715 \pm 0.014$ |
| 20 | $0.679 \pm 0.064$ | $0.679 \pm 0.063$ | $0.583 \pm 0.107$ | $0.643 \pm 0.114$ | $0.665 \pm 0.061$ | $0.704 \pm 0.021$ |

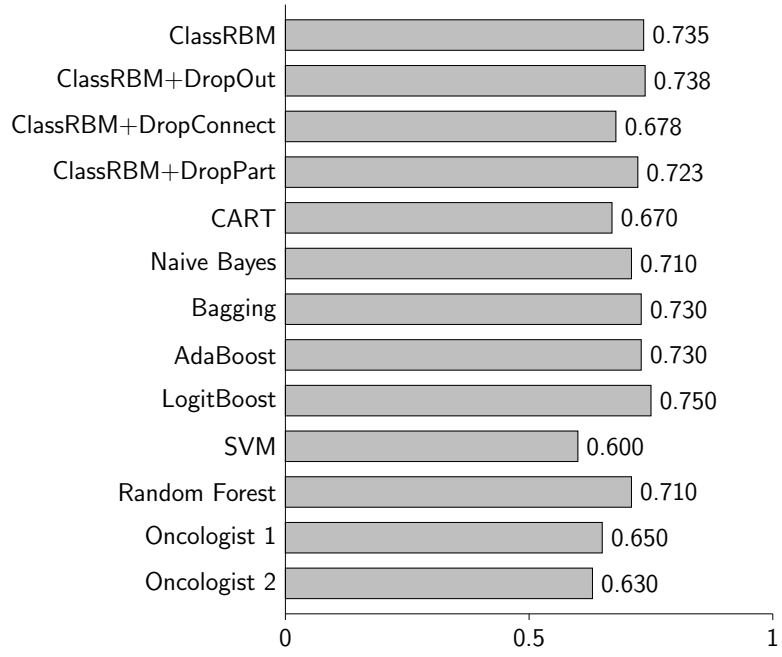


Figure 1: Classification accuracy of considered models and two human oncologists. Note that the results for oncologists were obtained using part of test set (100 cases only) [11].

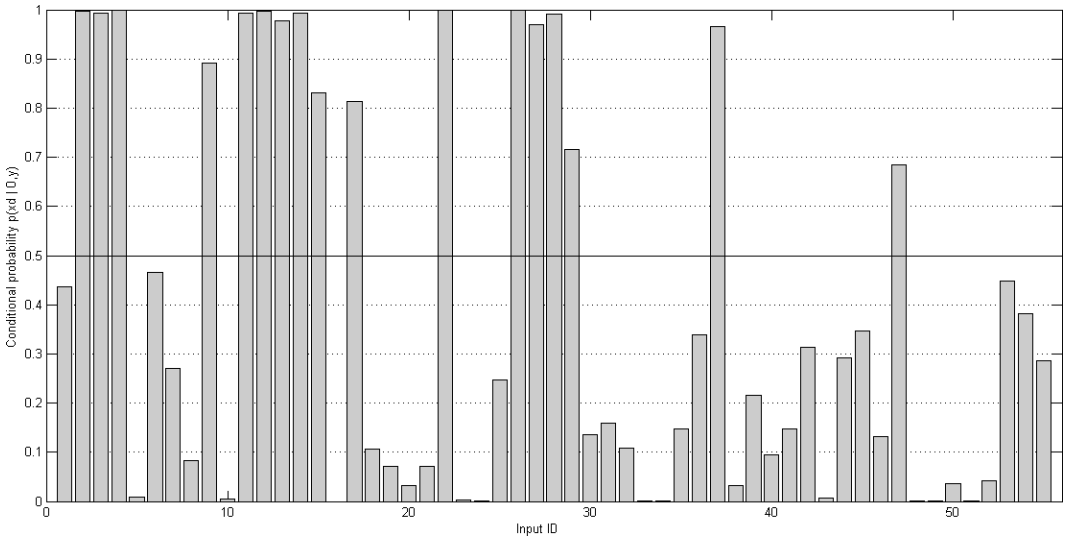


Figure 2: Discovered relevant inputs for ClassRBM with *DropPart*. The threshold is denoted as a solid horizontal line.

At the end, we have applied the ClassRBM with *DropPart* to discover relevant inputs (see Figure 2). Because we obtained probabilities of all inputs given the information about the breast cancer return and assuming absence of other inputs. For example, it turned out that inputs Nos. 11, 12, 13, i.e., histological type of tumor: ductual, lobular, and other, are important. Similarly, input No. 14, i.e., level of progesterone receptors in tumor (in fmol per mg of protein) less than 10, is also relevant. On the other hand, it seems to be quite surprising that tumor stage over 50 mm (input No. 5) does not matter in the breast cancer recurrence. However, medical doctors should determine now whether discovered features are interesting from medical or clinical point of view.

## 4 Conclusions

In this preliminary study we showed how to apply Classification Restricted Boltzmann Machine to the problem of predicting the breast cancer recurrence. We proposed the general probabilistic framework for learning with masks, called *Dropping*. The fashion of generating masks leads to different learning methods, i.e, *DropOut*, *DropConnect*, and a new method called *DropPart*. Our considerations are presented for ClassRBM but it can be straightforwardly applied to RBM, Deep Networks or Deep Boltzmann Machines.

In this paper, we do not address several important issues, e.g., how to perform prediction with masks (e.g. an approximate inference using Gaussian approximation is proposed in [13]), how to perform one learning iteration properly (see sum in Equation 14). Additionally, the performance of the methods can be improved by implementing some optimization tricks, e.g., mini-batch [2], centering trick [9]. Moreover, we have fixed the parameters values, however, we have an impression that in *DropConnect* and *DropPart* more learning iterations should be used in order to obtain more stable and thus better results. Last but not least, more thorough empirical studies are needed. We leave investigating all of these issues for further research.

### Acknowledgments

The research conducted by Jakub M. Tomczak has been partially co-financed by the European Union within the European Social Fund.

## References

[1] Polish National Cancer Registry. http://epid.coi.waw.pl/krn/english/index.asp.

[2] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[3] G. Hinton. A practical guide to training restricted Boltzmann machines. *Momentum*, 9(1), 2010.

[4] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[5] I. Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109, 2001.

[6] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th International Conference on Machine learning*, pages 536–543. ACM, 2008.

[7] H. Larochelle, M. Mandel, R. Pascanu, and Y. Bengio. Learning algorithms for the classification restricted Boltzmann machine. *The Journal of Machine Learning Research*, 13:643–669, 2012.

[8] B. M. Marlin, K. Swersky, B. Chen, and N. D. Freitas. Inductive principles for restricted Boltzmann machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 509–516, 2010.

[9] G. Montavon and K.-R. Müller. Deep boltzmann machines and the centering trick. In *Neural Networks: Tricks of the Trade*, pages 621–637. Springer, 2012.

[10] N. Srivastava. *Improving neural networks with dropout*. PhD thesis, University of Toronto, 2013.

[11] E. Štrumbelj, Z. Bosnić, I. Kononenko, B. Zakotnik, and C. G. Kuhar. Explanation and reliability of prediction models: the case of breast cancer recurrence. *Knowledge and information systems*, 24(2):305–324, 2010.

[12] J. M. Tomczak and A. Gonczarek. Decision rules extraction from data stream in the presence of changing context for diabetes treatment. *Knowledge and Information Systems*, 34(3):521–546, 2013.

[13] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of Neural Networks using DropConnect. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1058–1066, 2013.

# Appendix

Description of all binarized inputs:

1. menopausal status *false*,
2. menopausal status *true*,
3. Tumor stage less than 20 mm,
4. Tumor stage between 20 and 50 mm,
5. Tumor stage over 50 mm,
6. Tumor grade good,
7. Tumor grade medium,
8. Tumor grade poor,
9. Tumor grade not applicable,
10. Tumor grade not determined,
11. Histological type of the tumor ductal,
12. Histological type of the tumor lobular,
13. Histological type of the tumor other,
14. Level of progesterone receptors in tumor (in fmol per mg of protein) less than 10,
15. Level of progesterone receptors in tumor (in fmol per mg of protein) more than 10,
16. Level of progesterone receptors in tumor (in fmol per mg of protein) unknown,
17. Invasiveness of the tumor no,
18. Invasiveness of the tumor invades the skin,
19. Invasiveness of the tumor invades the mamilla,
20. Invasiveness of the tumor invades skin and mamilla,
21. Invasiveness of the tumor invades wall or muscle,
22. Number of involved lymph nodes 0,
23. Number of involved lymph nodes between 1 and 3,
24. Number of involved lymph nodes between 4 and 9,
25. Number of involved lymph nodes 10 or more,
26. application of a therapy (cTherapy) *false*,
27. application of a therapy (cTherapy) *true*,
28. application of a therapy (hTherapy) *false*,
29. application of a therapy (hTherapy) *true*,
30. Medical history no cancer,
31. Medical history 1st generation breast, ovarian or prostate cancer,
32. Medical history 2nd generation breast, ovarian or prostate cancer,
33. Medical history unknown gynecological cancer,
34. Medical history colon or pancreas cancer,
35. Medical history other or unknown cancers,
36. Medical history not determined,
37. lymphatic or vascular invasion *false*
38. lymphatic or vascular invasion *true*
39. Level of estrogen receptors in tumor (in fmol per mg of protein) less than 5,
40. Level of estrogen receptors in tumor (in fmol per mg of protein) 5 to 10,
41. Level of estrogen receptors in tumor (in fmol per mg of protein) 10 to 30,
42. Level of estrogen receptors in tumor (in fmol per mg of protein) more than 30,
43. Level of estrogen receptors in tumor (in fmol per mg of protein) not determined,
44. Diameter of the largest removed lymph node less than 15 mm,
45. Diameter of the largest removed lymph node between 15 and 20 mm,
46. Diameter of the largest removed lymph node more than 20 mm,
47. Ratio between involved and total lymph nodes removed 0,
48. Ratio between involved and total lymph nodes removed less that $10\%$,
49. Ratio between involved and total lymph nodes removed between 10 and $30\%$,
50. Ratio between involved and total lymph nodes removed over $30\%$,
51. Patient age group under 40,
52. Patient age group 40-50,
53. Patient age group 50-60,
54. Patient age group 60-70,
55. Patient age group over 70 years.