DOI https://doi.org/10.15407/csc.2023.03.005
UDC 004.932

V.M. KRYGIN, Junior Researcher, International Research and Training Centre for Information Technologies and Systems of the NAS and MES of Ukraine,
Glushkov ave., 40, Kyiv, 03187, Ukraine,
ORCID: https://orcid.org/0000-0002-9000-1685,
valeriy.krygin@gmail.com

USING GIBBS SAMPLING TO ESTIMATE THE SOLUTION OF THE UNPAIRED LEARNING PROBLEM

The article describes unpaired learning using Monte Carlo Markov Chain on the example of a stereo vision problem. The description includes the inference of the algorithm, the application of the stochastic gradient method, and some implementation details. Multiple penalty functions are considered, and quantitative results are presented. The results of the experiments expose new insights into weights for graphical models for stereo vision problems.

Keywords: *unpaired learning, Gibbs sampling, Monte Carlo Markov Chain, stereo vision.*

Introduction

In machine learning tasks, there are situations when it is difficult or impossible to obtain a training sample consisting of pairs of observations and corresponding hidden states. An example of such a task is the style transfer: we can collect examples of drawings by Ivan Aivazovsky and Salvador Dali; however, it is difficult to redraw the paintings of one of these artists as if another one drew them. In such cases, it may be possible to state the problem as the task of unpaired learning, the input data of which are two sets: a set of observations and a set of hidden states, and the observations may not correspond to the hidden states.

The first work devoted to unpaired learning is considered [1], where it is applied to image style transfer using random fields. One of the most cited works on this topic is [2], in which artificial neural networks are used.

This article, as well as the work [1], is dedicated to using the unpaired learning method to find parameters of a random field. The main differences are using of Gibbs sampling [3] to evaluate expected values and using binocular stereo vision problem for the experiments.

The structure of the current work is as follows: first, one of the possible tasks of unpaired learning is stated, then the proposed algorithm for its approximate solution is described, and then we provide the description and results of experiments.

Statement of the Problem of Unpaired Learning

This section provides basic definitions and one of the possible formulations of the problem of unpaired learning.

1. Random Field

We will call a grayscale image x that is $h \in \mathbb{N}_+$ pixels high and $w \in \mathbb{N}_+$ pixels wide a mapping from a set $T = \{1,...,h\} \times \{1,...,w\}$ of pixels into the set of colors $C = \{0,...,255\}$. The notation $x(y, j) = c$ means that the pixel located in row i and column j of image x has color $c \in C$. On the set of pixels, we define a non-empty neighborhood structure $\Gamma \subset 2^T$, $\gamma \in \Gamma \Rightarrow |\gamma| = 2$. Let tt' denote an unordered pair $\{t, t'\} \in \Gamma$ of neighboring objects and denote the set of all neighbors of an object t by $N_t = \{t' : tt' \in \Gamma\}$.

Let us define a finite non-empty set K of labels. The function k: $T \to K$ will be called labeling. The fact that the labeling k maps an object $t \in T$ to a label $\ell \in K$ will be denoted by $k_t = \ell$.

We call an ordered pair $(t, k) \in T \times K$ a vertex, and we call an unordered pair of $\{(t, k), (t', k')\}$ vertices, where $k \in K, k' \in K$, and $tt' \in \Gamma$, an edge. Let us introduce the function $q : C \times K \to \mathbb{R}$ of vertices' weights and the function $g : K^2 \to \mathbb{R}$ of edges' weights. Let k be a random field [4] with the distribution

$$p(k; g) = \frac{1}{G(g)} \cdot \exp\left\{\sum_{tt' \in \Gamma} g(k_t, k_{t'})\right\}, \qquad (1)$$

where $G(g)$ is the normalizing constant

$$G(g) = \sum_{k \in K^T} \exp\left\{\sum_{k \in K^T} g(k_t, k_{t'})\right\}.$$

Under known labeling k, the probability of image x mapping a pixel t to a value x_t equals

$$p(x_t | k; q) = \frac{1}{Q(k; q)} \cdot \exp\{q(x_t, k_t)\}, \qquad (2)$$

where $Q(k; q)$ is the normalizing constant

$$Q(k; q) = \sum_{x \in C^T} \exp\left\{\sum_{t \in T} q(x_t, k_t)\right\}.$$

If the colors of pixels are independently distribu ted, given the known labeling, image x has the distribution

$$p(x | k; q) = \frac{1}{Q(k; q)} \cdot \exp\left\{\sum_{t \in T} q(x_t, k_t)\right\}.$$

Then the pair (k, x) is a conditional random field [5] with the distribution

$$p(k, x; q, g) = p(x | k; q) \cdot p(k; g) = \frac{1}{Z(k, x; q, g)} \cdot$$

$$\cdot \exp\left\{\sum_{t \in T} q(x_t, k_t) + \sum_{tt' \in \Gamma} g(k_t, k_{t'})\right\}, \qquad (3)$$

where $Z(k, x; q, g) = G(g) \cdot Q(k; q)$ is the normalizing constant.

2. Statement of the Problem of Estimating Parameters of Random Fields

Let us have a set of labelings $\overline{k} = (k^1, \ldots, k^n)$; the weights q and g and the images corresponding to these labelings are unknown. We want to find the weights g by maximizing the log-likelihood function. Its partial derivatives with respect to $g(k, k')$ for all $k \in K, k' \in K$, are [6]

$$\frac{\partial}{\partial g(\ell, \ell')} \ln \prod_{i=1}^{n} p(k^i; g) = \sum_{i=1}^{n} \sum_{tt' \in \Gamma} \left[k_t^i = \ell \wedge k_{t'}^i = \ell'\right] -$$

$$-n \cdot \sum_{k \in K^T} p(k; g) \cdot \sum_{tt' \in \Gamma} [k_t = \ell \wedge k_{t'} = \ell'], \qquad (4)$$

that is, the difference between the empirical amount of the edges connecting specific labels and the expected frequency of these edges multiplied by n. Note that the log-likelihood function for such a parameterization is concave [6], which allows finding its maximum using convex optimization methods, and the existence of a gradient enables the use of gradient methods. The difficulty is that the computation of the expectation requires the summation of $|K|^{|T|}$ small numbers. To overcome this obstacle, we apply Gibbs sampling, which will be discussed in the next subsection.

Now we want to estimate the weights q using the maximum likelihood method given the set of images $\overline{x} = (x^1, x^2, \ldots, x^n)$. To achieve this goal, we decided to modify the EM algorithm of unsupervised learning [7]. Let us begin with the equality

$$\ln \prod_{i=1}^{n} p(x^i; q^j) = \sum_{i=1}^{n} \sum_{k \in K^T} \frac{p(x^i, k; q^j, g)}{\sum_{k'} p(x^i, k'; q^j, g)} \cdot$$

$$\cdot\left[\ln p(k; g) + \ln p(x^i | k; q^j) - \ln p(k | x^i; q^j, g)\right].$$

We use the g that was found with the help of empirical material $\overline{k}$ by maximizing the log-likelihood function. Therefore, the summand

$$\sum_{i=1}^{n} \sum_{k \in K^T} \frac{p(x^i, k; q^j, g)}{\sum_{k' \in K^T} p(x^i, k'; q^j, g)} \cdot \ln p(k; g)$$

will stay constant. To increase the second term, it is enough to find such q^{j+1} that

$$\sum_{i=1}^{n} \sum_{k \in K^T} \frac{p(x^i, k; q^j, g)}{\sum_{k' \in K^T} p(x^i, k'; q^j, g)} \cdot \ln p(x^i | k; q^j) <$$

$$< \sum_{i=1}^{n} \sum_{k \in K^T} \frac{p(x^i, k; q^j, g)}{\sum_{k' \in K^T} p(x^i, k'; q^j, g)} \cdot \ln p(x^i | k; q^{j+1}), \qquad (5)$$

if it exists. If it does not exist, the problem is solved. Shannon's lemma implies [8]

$$-\frac{p(x^i,k;q^j,g)}{\sum_{k'\in K^T} p(x^i,k;q^j,g)}\ln p(k|x^i;q^j,g)\le$$
$$\le -\frac{p(x^i,k;q^j,g)}{\sum_{k'\in K^T} p(x^i,k;q^j,g)}\ln p(k|x^i;q^{j+1},g),$$

meaning that the third summand will not decrease after replacing q^j with q^{j+1} within the logarithm. Hence, we have

$$\ln\prod_{i=1}^{n} p(x^i;q^j) < \ln\prod_{i=1}^{n} p(x^i;q^{j+1}).$$

Therefore, one can perform step-by-step search for new q^{j+1}, $j\in\mathbb{N}_+$, and thus improve the value of the likelihood function. The value of q^0 can be chosen arbitrarily (for example, $q^0\equiv 0$).

Let us consider a partial case of the problem (5) as the maximization problem

$$q^{j+1}\in\arg\max_{f:C\times K\to\mathbb{R}}\sum_{i=1}^{n}\sum_{k\in K^T}\frac{p(x^i,k;q^j,g)}{\sum_{k'\in K^T} p(x^i,k';q^j,g)}\cdot$$
$$\cdot\ln p(x^i|k;f). \tag{6}$$

For convenience, let

$$p(k|x^i,k;q^j,g)=\frac{p(x^i,k;q^j,g)}{\sum_{k'\in K^T} p(x^i,k';q^j,g)}. \tag{7}$$

We need a gradient of the expression that needs to be maximized in (6). The partial derivative of its i-th summand with respect to $f(c,\ell)$ for each $c\in C$ and $\ell\in K$ is calculated by the formula

$$\frac{\partial}{\partial f(c,\ell)}\sum_{k\in K^T} p(k|x^i;q^j,g)\cdot\ln p(x^i|k;f)=$$
$$=\sum_{k\in K^T} p(k|x^i;q^j,g)\cdot u^i(k), \tag{8}$$

where

$$u^i(k)=\sum_{t\in T}[x_t^i=c\wedge k_t=\ell]-|T|\cdot$$
$$\cdot\sum_{c'\in C} p(c'|k;f)\cdot[c'=c\wedge k_t=\ell].$$

Finding q and g using the maximum likelihood estimation is an intractable optimization problem in general case because it requires the summation of a vast number of small numbers. However, both functions are concave, which theoretically allows gradient optimization. In the next part of the work, stochastic gradient descent is described, which allows us to partially overcome the described difficulties and obtain acceptable results in practice.

Let us formulate the problem of unpaired learning of parameters of a random field in the context of this work. We have sets T of pixels, $\Gamma\subset 2^T$ of neighbors, K of labels, and C of colors. We know that images x and corresponding labelings k have a statistical relationship, which is described by the formula (3), labelings have a distribution of the form (1), and the probability of an image to map a pixel to a specific color given the known labeling has the form (2). The input is a collection $\bar{k}$ of labelings and a collection $\bar{x}$ of images, which do not have to correspond to each other. We need to find estimates of parameters q and g.

Finding Estimates of Parameters of Random Fields

Calculating gradients of logarithms of probability functions of random fields is an intractable problem because it requires the summation of too many small numbers. Therefore, we use a stochastic gradient – random variable, expected value of which equals the actual value of the gradient. Since the gradients we are interested in are mathematical expectations of certain functions, the Monte Carlo method can be applied.

1. Stochastic Gradient Method

We will sketch the application of the stochastic gradient method to tasks described in the current work. Let values from a finite set U be given probabilities p: $U\to[0;\ 1]$, and suppose the gradient u of some concave continuous function, which we want to maximize, can be represented as an expectation

$$u(s)=\sum_{s'\in U} p(s')\cdot v(s,s'), \tag{9}$$

where v is some known function. Let us apply the Monte Carlo method to calculate the estimate $\tilde{u}(s)$ of the gradient u at the point $s\in U$: we generate m independent values $\bar{s}=(s^1,\ldots,s^m)$ with the distribution p and calculate their average value

$$\tilde{u}(s)=\sum_{j=1}^{m}\frac{v(s,s^j)}{m}.$$

We apply this estimate as in the usual gradient method: given such a sequence $a^i\in\mathbb{R}_+$, $i\in\mathbb{N}_+$ that

$$\lim_{i\to\infty} a^i = 0,$$

$$\lim_{i\to\infty} \sum_{i=0}^{n} a^i = \infty,$$

we choose an arbitrary $s^0 \in U$ and compute [9]

$$s^i = s^{i-1} + a^j \cdot \tilde{u}(s^{i-1}) \quad (10)$$

for any step $i \in \mathbb{N}_+$. Choosing the number of steps is out of the scope of this paper.

2. Gibbs Sampling

We need to generate values from a known distribution to estimate the gradient. In our case, this task is difficult because images can consist of thousands, millions, or hundreds of millions of pixels. To overcome this problem, we use Gibbs sampling [3]. Let us consider its application in the example of labeling sampling with known distribution (1) $p(k;g) = \frac{1}{G(g)} \cdot \exp\left\{\sum_{tt'\in\Gamma} g(k_t, k_{t'})\right\}$, for a known mapping g. Let us choose an arbitrary labeling $k: T \to K$ and an arbitrary pixel $t \in T$. Hence the labels in all neighboring pixels are known, it is possible to calculate the distribution of a label:

$$p(k_t \,|\, k_{t'},\ t' \in N_t; g) = \frac{\exp\left\{\sum_{t'\in N_t} g(k_t, k_{t'})\right\}}{\sum_{\ell\in K} \exp\left\{\sum_{t'\in N_t} g(\ell, k_{t'})\right\}}.$$

Sample a label from this distribution and write the value to k_t. The procedure of calculating the distribution, generating the label, and modifying the labeling should be performed for each pixel $t \in T$. We will call this sequence of actions the iteration of Gibbs sampling. One needs to use the labeling k obtained after the previous iteration at each iteration.

3. Application of Stochastic Gradient Method

To find g, we need to generate random labelings with known probabilities $p(k; g^j)$, $j \in \mathbb{N}_0$,, where g^j is the value of parameter g at step j of the stochastic gradient method, to evaluate the partial derivatives (4).

To solve problem (6) we need to evaluate partial derivatives (8) for a known collection $\overline{x}$, known mapping g (which is the solution to the problem (5)), and known mapping q. This requires sampling of random labelings with probabilities $p(k \,|\, x^i; q^j, g)$. In this paper, we consider the case $\overline{x} = (x^1)$, that is, recognition of a single image x^1 given a collection $\overline{k}$. Only f varies when we solve the problem (6). Hence, if there is only one image x^1, the multiplier $\sum_{k'\in K^T} p(x^1, k', q^j, g)$ can be carried out from within the argmax because it does not depend on f and it does not affect the result. Now, to find q^{j+1}, $j \in \mathbb{N}_0$, it is enough to solve the problem $q^{j+1} \in \in \max_{f: C\times K\to\mathbb{R}} \sum_{k\in K^T} p(k, x^1; q^j, g) \cdot \ln p(x^1 \,|\, k; f)$ approximately using the stochastic gradient method (10).

Binocular Stereo vision Problem

The section is devoted to several possible formulations of the problem of binocular stereo vision [10] and experimental checks of the described approach to unpaired learning.

1. Bayesian Recognition Problem

Bayesian recognition problem [11; 8] is finding such an answer d from the finite set D^T of possible answers for the input signal x that for a given loss function $w: D^T \times K^T \to \mathbb{R}$ is the solution of the optimization problem

$$d^* \in \arg\min_{d:T\to D} \sum_{d:T\to D} w(d,k) \cdot p(x,k). \quad (11)$$

If $D = K$ and $w(d, d') = [d \neq d']$, we have a problem $d^* \in \arg\min_{d:T\to K} \sum_{k:T\to K} p(x,k) \cdot [d \neq k] = \max_{d:T\to K} p(x,d)$.

To state a more relevant problem, consider a case $D = K$ and

$$w(d, d') = \sum_{t\in T} [d_T \neq d'_T]. \quad (12)$$

The solution to the problem with such loss function requires the calculation of marginal distributions of labels for each pixel [12]

$$\begin{aligned} d^* &\in \arg\min_{d:T\to K} \sum_{k:T\to K} p(x,k) \cdot \sum_{t\in T} [d_t = k_t] = \\ &= \arg\max_{d:T\to K} \sum_{t\in T} \sum_{\substack{k:T\to K \\ k_t = d_t}} p(x,k) \Rightarrow \\ &\Rightarrow d_t^* \in \arg\max_{d:T\to K} \sum_{\substack{k:T\to K \\ k_t = d_t}} p(x,k). \end{aligned} \quad (13)$$

Finding marginal probabilities of labels $k \in K$ for each pixel $t \in T$ requires computation of $T| \cdot |K|$ sums, each containing $|K|^{|T|}$ numbers.

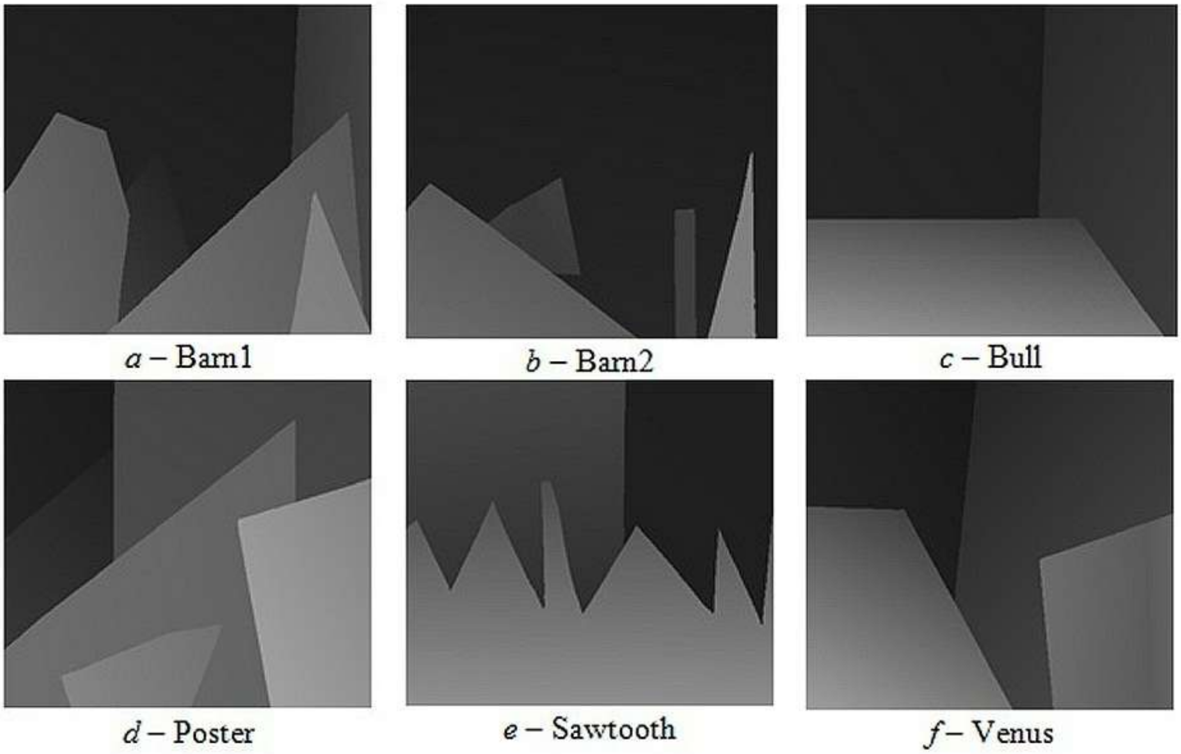


Fig. 1. Images of depth maps from the dataset [14], which were used to estimate g

Let us look at one more loss function because it leads to interesting results: for the function

$$w(d,k) = (d-k)^2, \qquad (14)$$

if D is a convex closure of the set K (and thus continuous), the solution of the Bayesian recognition problem is the conditional expectation [12; 8]

$$d^* \in \arg\min_{d:T\to D} \sum_{k:T\to K} p(x,k)\cdot\sum_{t\in T}(d_t-k_t)^2 \Rightarrow$$

$$\Rightarrow d_t^* = \frac{\sum_{k:T\to K} p(x,k)\cdot k_t}{\sum_{k:T\to K} p(x,k)}. \qquad (15)$$

2. Experimental Results

In the experiments, we used a computer with a central processor Intel Core i7-8550U (4 cores, 8 threads, clock frequency 1,8 GHz) and a graphics processor Nvidia GeForce MX150 (maximum clock frequency 1,5 GHz, 3 computing units, 2 GB of RAM). Parallel computing on graphics processors was used with single-precision floating-point numbers (32-bit float) and CUDA technology [13].

Let us define two images $x_1 : T \to C$ and $x_2 : T \to C$ finite se $K \subset \{0\}\times\mathbb{Z}$ of possible labels (we will also call them disparities) and neighborhood structure $\Gamma = \left\{\{t,t'\}\in 2^T : \|t-t'\|_{L_1} = 1\right\}$.

For $\theta\in\mathbb{R}$, let weights g be given by $g(\ell,\ell',\theta) = 1+\theta\cdot[\ell\neq\ell']$.

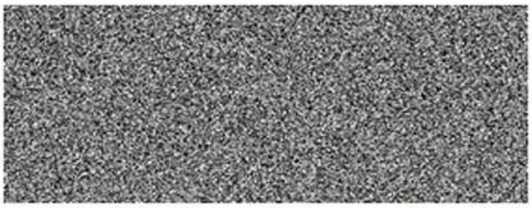
For a certain $m\in\mathbb{N}_+$ and the finite set $\overline{\beta}=(\beta_0,\ldots\beta_{m+1})$ let weights q be defined by $q(x_t,k_t;\overline{\beta}) = \sum_{j=0}^{m}\beta_j\cdot$

$$\cdot\left[\,|x_1(t)-x_2(t+k_t)| = j\,\right] +$$

$$+\beta_{m+1}\cdot\left[\,|x_1(t)-x_2(t+k_t)| > m\,\right],$$

that is, we assign a weight for each difference between the intensities of the corresponding pixels $x_1(t)$ and $x_2(t+k_t)$. A separate weight β_{m+1} is used for the differences exceeding m.

The value of g was evaluated from data provided by Middlebury College [14]. We used the following scenes to take labelings from: Barn1 (Fig. 1,*a*), Barn2 (Fig. 1,*b*), Bull (Fig. 1,*c*), Poster (Fig. 1,*d*), Sawtooth (Fig. 1,*e*), and Venus (Fig. 1,*f*).



a – First iteration: the image of a white noise

b – Second iteration: different labels are grouped, but the changes are quite frequent: $1{,}8 \cdot 10^4$ transitions instead of the required $5{,}2 \cdot 10^3$

c – Iteration 42: a lot of black pixels (label 0) because pixels in the images usually correspond to the objects with the same distance from the camera

d – Iteration 47 (last)

Fig. 2. Examples of generated labelings *g*

a – Left image

b – Right image

c – Disparity map

Fig. 3. Tsukuba Stereo pair from the dataset [14]

The dataset contains real-valued labelings with an accuracy of 1/8 part of a pixel, so the values were rounded, which reduced the number of labels in the set K to 21. The computation of parameters g was performed using 47 steps of the stochastic gradient method (10) with steps' lengths

$$a^i = \frac{1}{\sqrt{i} \cdot |T|},\ i = 1,\ a^i = \frac{10}{\sqrt{i} \cdot |T|},\ i \geq 2.$$

For each iteration i of the gradient method, we used $100+i$ average values for applying the Monte Carlo method. We sampled $w+h+10 \cdot i$ images of size w=500 by h=200 pixels to compute each average value. The procedure took six hours for 47 iterations. The result is $\theta \approx -34{,}54$.

Note that hundredths matter. Therefore, rounding to tenths was unacceptable. For example, $\theta \approx -34{,}63$, black images were generated (relative error of 100), and at $\theta \approx -34{,}59$, the results were almost perfect (relative error of 0,5 %). Visualization of some labelings is shown in Fig. 2. There are a lot of black pixels, which correspond to the label 0. Since the form of the function g does not depend on the specific values of the neighboring labels, we are only interested in the amount of neighboring pixels that have different labels.

a – Maximization of marginal distributions: failure rate is 48%, MSE is 15,83

b – Average: failure rate is 82%, MSE is 4,84

c – One of the sampled images

Fig. 4. Second iteration: outlines of some objects are visible

a – Maximization of marginal distributions: failure rate is 21%, MSE is 3,49

b – Average: failure rate is 59%, MSE is 3,10

c – One of the sampled images

Fig. 5. Third iteration: objects are much cleaner

a – Maximization of marginal distributions: failure rate is 14%, MSE is 1,98

b – Average: failure rate is 39%, MSE is 2,04

c – One of the sampled images

Fig. 6. Iteration 89: the disparity map is ready except for a few segments

We used the Tsukuba stereo pair (Fig. 3) to estimate q. The number of labels is $|K| = 21$, as for the evaluation of g. The parameter β contains 64 elements. The computation of q was performed using 88 steps of the stochastic gradient method (10) with steps' lengths $a^i = \frac{10}{\lceil (i-1)/10 \rceil + 1}$, $i \in \mathbb{N}_+$.

For each iteration of the gradient method, the Gibbs sampler has generated $\lceil (w+h)/2 \rceil + i$ images of size w=384 by h=288 pixels to obtain 100+i average values for applying the Monte Carlo method.

The first iteration took 102 seconds and generated a black image, the second lasted 177 seconds

a – Maximization of marginal distributions: failure rate is 15%, MSE is 2,06 *b* – Average: failure rate is 38%, MSE is 2,02 *c*– One of the sampled images

Fig. 7. Iteration 125: no significant changes are observed

(Fig. 4), and the third took 200 seconds (Fig. 5). The entire procedure took almost 15 hours for 89 iterations (Fig. 6). Additionally, 36 iterations were performed for 12,5 hours. These iterations slightly improved the result visually (Fig. 7). In the figures 4–7, one can see the visualization of maximization of estimates of marginal distributions for each pixel (13), estimates of expected values of labels for each pixel (15), and one of the generated images that were used to construct the corresponding estimates. The figures show that the sampling results (Fig. 4,*c*, 5,*c*, 6,*c*, and 7,*c*) look noisy. However, an image of the most frequent labels in each pixel (Fig. 4,*a*, 5,*a*, 6,*a*, and 7,*a*) formed a fairly clear idea of distances to objects. Visualizations of average values (Fig. 4,*b*, 5,*b*, 6,*b*, and 7,*b*) look blurry. However, some objects can also be seen on them.

For quantitative comparison of the results, we use average values of loss functions (12) and (14), which we call failure rate and MSE (mean squared error) respectively. To calculate the failure rate for the average image, we round the values of the average image. We ignore the regions that correspond to black pixels on the ground truth disparity map (Fig. 3,*c*) for a fair comparison because they indicate missing information. In our experiments, maximizing marginal probabilities produced labelings with lower failure rates than the average images. MSE is lower for the average images on the early iterations. This corresponds to the penalty functions used.

The value of the parameter $\overline{\beta}$ is shown graphically in Fig. 8. Note that β_i often has the form $\beta_i = -b \cdot i + c$ or $\beta_i = -b \cdot i^2 + c$ for certain $b \in \mathbb{R}$ and $c \in \mathbb{R}$. It can be seen from the graph that the extremum is not at point 0, but at point 1, and even the value of β_2 is greater than the value of β_0. This contradicts the widespread practice of using squared difference or absolute value of intensities as β_i. The estimate of parameter β_{63}, which matches all color differences that are greater than 62, is -5,56.

Conclusions

Unpaired learning of parameters of random fields is a theoretically grounded and practically helpful approach in pattern recognition. With the development of parallel computing on GPUs, Gibbs sampling becomes an increasingly attractive tool for working with random fields.

The experiments showed results that were previously unknown to the authors: function (Fig. 8) of vertex weights may not have the maximum weight at the point where the color difference of the corresponding pixels is zero, and at the point where the pixels are slightly different. It can be interpreted as follows: the camera receives images slightly distorted by noise and discreteness of the sensor perception, so the probability that the corresponding pixels of two images of a stereo pair have the same color is lower than the probability that their colors are slightly different. This is a fascinating fact for further research.

Also, the experiments showed us fairly expected results: maximizing marginal probabilities pro-

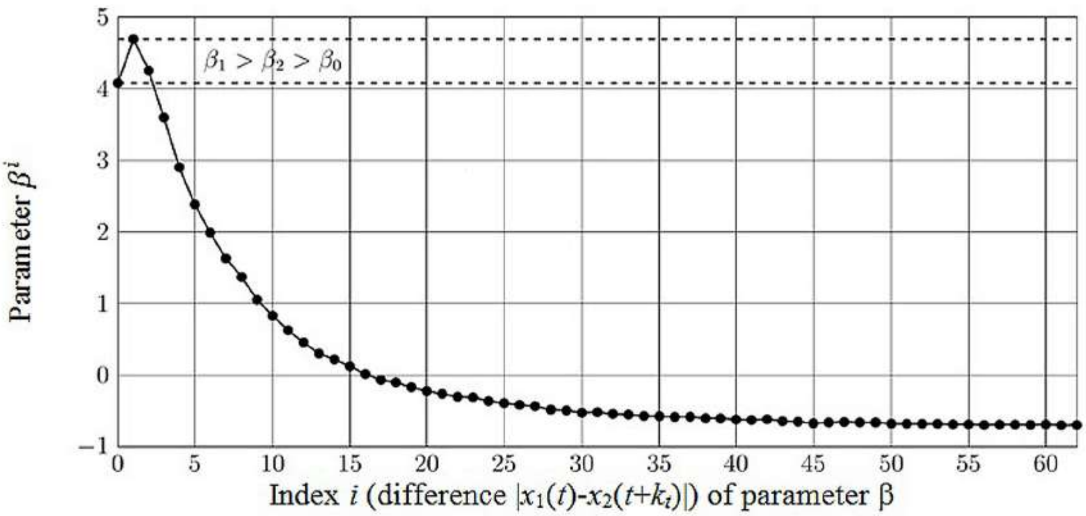


Fig. 8. The plot of β

duced labeling with lower failure rates than the average images because this approach was designed to solve a Bayesian problem with this penalty function. Though, the average images do not always have lower MSE. A lower convergence speed of averaging may explain this compared to maximizing marginal probabilities. One of the essential drawbacks of the used approaches is their operation time. Therefore, using them in combination with fast algorithms may be feasible. For example, the result of training (mappings q and g) can be submitted to the input of an algorithm that iteratively calculates the approximation to the solution of the problems (11) (for example, [15—17]).

REFERENCES

1. Resales, R., Achan, K., Frey, B., 2003. "Unsupervised image translation". Proc. of Ninth IEEE International Conference of Computer Vision, 2003. Vol. 4. pp. 472—478. DOI: 10.1109/ICCV.2003.1238384.
2. Zhu, J.-Y. et al., 2017. "Unpaired image-to-image translation using cycle-consistent adversarial networks". Proceedings of the IEEE international conference on computer vision, pp. 2223—2232.
3. Geman, S., Geman, D, 1984. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". IEEE Transactions on pattern analysis and machine intelligence, 6, pp. 721—741.
4. Kindermann, R., Snell, L., 1980. "Markov random fields and their applications". American Mathematical Society, Vol. 1, pp. 4-7. DOI: 10.1090/conm/001.
5. Lafferty, J.D., McCallum, A., Pereira, F.C.N., 2001. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". Proceedings of the eighteenth international conference on machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 282—289.
6. Descombes, X. et al., 1999. "Estimation of Markov random field prior parameters using Markov chain Monte Carlo maximum Likelihood". IEEE Transactions on Image Processing, Vol. 8, n 7, pp. 954—963.
7. Shlezinger, M.I., 1968. "The interaction of learning and self-organization in pattern recognition". Cybernetics. Vol. 4, N 2, pp. 66—71.
8. Schlesinger, M, Hlavac, V., 2002. Ten Lectures on Statistical and Structural Pattern Recognition. Computational Imaging and Vision, Vol. 24. Kluwer Academic Publishers – Dordrecht/Boston/London, 520 p. [Шлезингер М.И., Главач В. Десять лекций по статистическому и структурному распознаванию. Киев: Наукова думка, 2004].
9. Robbins, H., Monro, S., 1951. "A Stochastic Approximation Method". The Annals of Mathematical Statistics. Institute of Mathematical Statistics, Vol. 22, N 3, pp. 400—407.

10. Szeliski, R, 2022. Computer vision - algorithms and applications. 2nd ed. 2022 Edition. Springer. 1232 p. ISBN 978-3030343712.
11. Duda, R.O., Hart, P.E., 1973. "Pattern classification and scene analysis". Wiley, pp. I–XVII, 1–482.
12. Schlesinger, D., 2003. "Gibbs probability distributions for stereo reconstruction". Pattern recognition / ed. Michaelis B., Krell G. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 394–401.
13. Nickolls, J., Buck, I., Garland, M., Skadron, K., 2008. Scalable parallel programming with CUDA: Is Cuda the parallel programming model that application developers have been waiting for? Queue, 6(2), pp. 40-53. DOI: 10.1145/1365490.1365500.
14. Scharstein, D., Szeliski, R., 2002. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". International Journal of Computer Vision. Kluwer Academic Publishers, Vol. 47, N 1-3, pp. 7–42.
15. Boykov, Y., Kolmogorov, V., 2004. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". IEEE Trans. Pattern Anal. Mach. Intell. USA: IEEE Computer Society, Vol. 26, N 9, pp. 1124–1137.
16. Kolmogorov, V., 2005. "Convergent tree-reweighted message passing for energy minimization". Interna-tional workshop on artificial intelligence and statistics PMLR, pp. 182–189.
17. Schlesinger, M.I., Antoniuk, K.V., 2011."Diffusion algorithms and structural recognition optimization problems," Cybern. Syst. Analysis, Vol. 47, No. 2, pp. 175–192. https://doi.org/10.1007/s10559-011-9300-z (In Russian) [Шлезингер М.И., Антонюк К.В. Анализ алгоритмов диффузии для решения оптимизационных задач структурного распознавания. Кибернетика и системный анализ. 2011. Т. 47, № 2. С. 3–20].

Надійшла 28.07.2023

В.М. Кригін, м.н.с., Міжнародний науково-навчальний центр
інформаційних технологій і систем НАН та МОН України,
03187, м. Київ, просп. Академіка Глушкова, 40, Україна.
ORCID: https://orcid.org/0000-0002-9000-1685,
valeriy.krygin@gmail.com

ВИКОРИСТАННЯ СЕМПЛЮВАННЯ ЗА ҐІББЗОМ ДЛЯ ОЦІНКИ РОЗВ'ЯЗКУ ЗАДАЧІ НЕСПАРЕНОГО НАВЧАННЯ

Вступ. У задачах машинного навчання виникають ситуації, коли складно або неможливо отримати навчальну вибірку, що складається з пар спостережень і відповідних їм прихованих станів досліджуваних об'єктів. Прикладом такої задачі є задача переносу стилю одного малюнку на інший: ми можемо зібрати приклади малюнків Івана Айвазовського та Сальвадора Далі, проте складно перемалювати картини одного з цих митців у стилі іншого. У таких випадках є можливість поставити задачу неспареного навчання, вхідними даними якої слугують два набори: набір спостережень і набір прихованих станів, причому спостереження можуть не відповідати станам.

Мета. Метою роботи є побудова та експериментальна перевірка алгоритму неспареного навчання, особливістю якого є використання семплювання за Ґіббзом. Такий підхід надає широкі можливості для паралельних обчислень, що дає змогу здійснювати експериментальну перевірку за допомогою технології *CUDA*, яка уможливлює виконання паралельних обчислень на графічних процесорах.

Результати. Експерименти візуально показали доцільність запропонованого підходу та надали чисельні показники, які вказують на коректність використання наведених методів мінімізації математичних очікувань певних штрафів, таких як сума квадратів відхилень і кількість невірно розпізнаних пікселів. Також в результаті неспареного навчання було виявлено, що для наведених даних функція вагів, яка відповідає інтенсивності шуму на зображенні, досягає оптимуму не в нулі, а в одиниці.

Методи. Для розв'язання задачі було використано семплювання за Ґіббзом і стохастичний градієнтний метод. Для експериментальної перевірки було використано технологію *CUDA* паралельних обчислень на графічних процесорах.

Висновки. Експерименти показали доцільність використання семплювання за Ґіббзом у задачах неспареного навчання параметрів графових моделей. Поточна реалізація потребує багато часу для оброблення зображення, але має значний потенціал для паралельних обчислень на *CPU, GPU, FPGA* або іншому пристрої чи мережі пристроїв, що дозволить значно прискорити її з розвитком обчислювальної техніки.

Ключові слова: *неспарене навчання, семплювання за Ґіббзом, Монте-Карло ланцюг Маркова, стереобачення.*