Bayesian multimodeling: variational inference

MIPT

2023

Variational calculus

Variational calculus problem is to find maxima and minima of functionals: mappings from a set of functions to the real numbers.

Example

Find a PDF p that gives maximum of entropy $H = -\int_w \log p(w) p(w) dw$.

- p — function to find
- H — functional

If a function is set from a predefined set of functions, we can consider the variational calculus problem as an approximation problem.

Model selection: coherent Bayesian inference

First level: find optimal parameters:

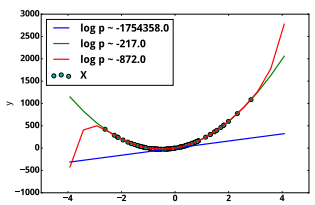
$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

Second level: find optimal model:

Evidence:

$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$

Model selection scheme



Polynomial regression example

Local variational optimization, idea

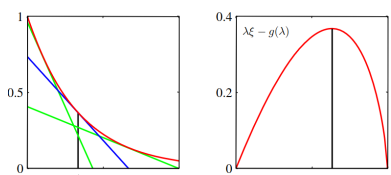
Consider a problem of approximation of $f(x) = \exp(-x)$ by linear function.
Any linear function will be a lower bound on $f(x)$ if it corresponds to a tangent. Use Taylor series:

$$y(x) = f(x_0) + f'(x_0)(x - x_0)$$

or, using $\lambda = -f(x_0)$

$$y(x) = \lambda x - \lambda + \lambda \log(-\lambda).$$

Where is the variational optimization here?



Local variational optimization, idea

Consider a problem of approximation of $f(x) = \exp(-x)$ by linear function.

Any linear function will be a lower bound on $f(x)$ if it corresponds to a tangent. Use Taylor series:

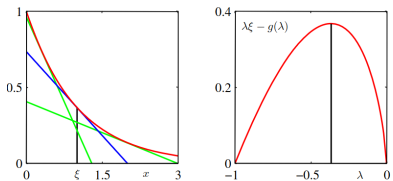
$$y(x) = f(x_0) + f'(x_0)(x - x_0)$$

or, using $\lambda = -f(x_0)$

$$y(x) = \lambda x - \lambda + \lambda \log(-\lambda).$$

We must find the tightest bound:

$$\max_{\lambda} \lambda x - \lambda + \lambda \log(-\lambda)$$



Local variational optimization and Evidence

Using similar approach we can approximate more interesting functions, for example sigmoid :

$$\log \sigma(x) = -\frac{x}{2} - \log(e^{\frac{x}{2}} + e^{\frac{-x}{2}}).$$

Note that $f(x) = -\log(e^{\frac{x}{2}} + e^{\frac{-x}{2}})$ is convex. Its approximation is:

$$\max_{x^2} \left(\lambda x^2 - f\left(\sqrt{x^2}\right)\right).$$

Optimal value gives:

$$\sigma(x) \geq \sigma(x_0)\exp\left((x - x_0)/2 - \lambda(x_0)\left(x_0^2 - x^2\right)\right).$$

The evidence integral becomes quadratic => we can use an approximation by Gaussian, similar to Laplace approximation.

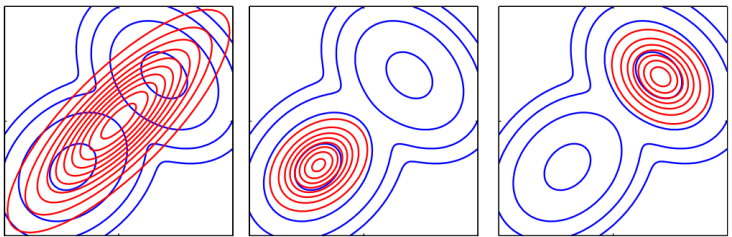
Evidence lower bound, ELBO

Evidence lower bound is a method of approximation of intractable distribution $p(\mathbf{w}|\mathfrak{D},\mathbf{h})$ with a distribution $q(\mathbf{w}) \in \mathfrak{Q}$.

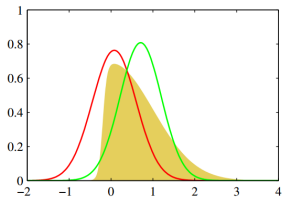
Evidence lower bound estimation often reduces to optimization problem

$$\log\ p(\mathfrak{D}|\mathbf{h}) \geq$$

$$\geq \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathfrak{D})) = -\int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{w}|\mathfrak{D})}{q(\mathbf{w})} d\mathbf{w} = \mathrm{E}_{\mathbf{w}} \log\ p(\mathfrak{D}|\mathbf{w}) - \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})).$$



Variational inference vs. expectation propogation (Bishop)



Laplace Approximation vs Variational inference

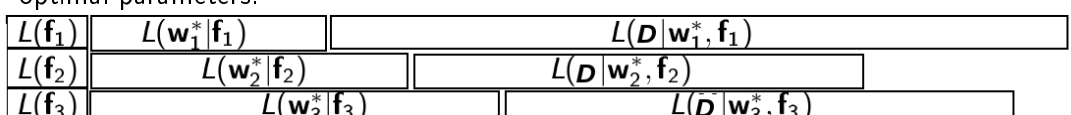
Minimum description length principle

$$\mathrm{MDL}(\mathbf{f}, \mathfrak{D}) = L(\mathbf{f}) + L(\mathfrak{D}|\mathbf{f}),$$

where $\mathbf{f}$ is a model, $\mathfrak{D}$ is a dataset, L is a description length in bits.

$$\mathrm{MDL}(\mathbf{f}, \mathfrak{D}) \sim L(\mathbf{f}) + L(\mathbf{w}^*|\mathbf{f}) + L(\mathfrak{D}|\mathbf{w}^*, \mathbf{f}),$$

$\mathbf{w}^*$ — optimal parameters.



ELBO estimation

ELBO maximization

$$\int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h})}{q(\mathbf{w})} d\mathbf{w}$$

is equivalent to KL-divergence minimization between $q(\mathbf{w}) \in \mathfrak{Q}$ and posterior distribution $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$:

$$\hat{q} = \arg\max_{q \in \mathfrak{Q}} \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathbf{y}, \mathbf{w}|\mathbf{X}, \mathbf{h})}{q(\mathbf{w})} d\mathbf{w} \Leftrightarrow$$

$$\hat{q} = \arg\min_{q \in \mathfrak{Q}} \mathrm{D}_{\mathrm{KL}}\left(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})\right),$$

$$\mathrm{D}_{\mathrm{KL}}\left(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})\right) = \int_{\mathbf{w}} q(\mathbf{w}) \log \left(\frac{q(\mathbf{w})}{p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})}\right) d\mathbf{w}.$$

ELBO and sample size

Statement

Let $m \gg 0$, $\lambda > 0$, $\frac{m}{\lambda} \in \mathbb{N}$, $\frac{m}{\lambda} \gg 0$. Then optimization

$$\mathsf{E}_q \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) - \lambda \mathsf{D}_{\mathsf{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})\big)$$

is equivalent to optimization of ELBO for a random subsample $\hat{\mathbf{y}}, \hat{\mathbf{X}}$ with size $\frac{m}{\lambda}$.

See also, [β-VAE, Fixing Broken ELBO, Talk from Olga Grebenkova, 2021].

What's better: ELBO or Laplace approximation?

(or MCMC)?

ELBO usage

ELBO: when to use?

- Evidence estimation;
- Latent distribution estimation (topic modeling, dimmension reduction).

Why ELBO?

- reduces the problem of ELBO estimation to optimization;
- scales easily (compare with Laplace approximati;
- easy to use in comparison to MC-based methods.

ELBO can give a very biased evidence estimation.

ELBO: normal distribution

Let $q \sim \mathcal{N}(\boldsymbol{\mu}_q, \mathbf{A}_q)$.
Then ELBO equals to:

$$\int_{\mathbf{w}} q(\mathbf{w}) \log\, p(\mathbf{Y}|\mathbf{X}, \mathbf{w}, \mathbf{h}) d\mathbf{w} - D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})\big) \simeq$$

$$\sum_{i=1}^{m} \log\, p(\mathbf{y}_i|\mathbf{x}_i, \hat{\mathbf{w}}) - D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})\big) \to \max_{\mathbf{A}_q, \boldsymbol{\mu}_q}, \quad \hat{\mathbf{w}} \sim q.$$

If prior $p(\mathbf{w}|\mathbf{h})$ is normal:

$$p(\mathbf{w}|\mathbf{h}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{A}),$$

KL-divergence $D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})\big)$ is computed analytically:

$$D_{\mathrm{KL}}\big(q(\mathbf{w})||p(\mathbf{w}|\mathbf{h})\big) = \frac{1}{2}\big(\mathrm{tr}(\mathbf{A}^{-1}\mathbf{A}_q) + (\boldsymbol{\mu} - \boldsymbol{\mu}_q)^{\mathsf{T}}\mathbf{A}^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_q) - n + \ln\,|\mathbf{A}| - \ln\,|\mathbf{A}_q|\big).$$

Graves, 2011

Prior: $p(\mathbf{w}|\sigma) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma\mathbf{I})$.

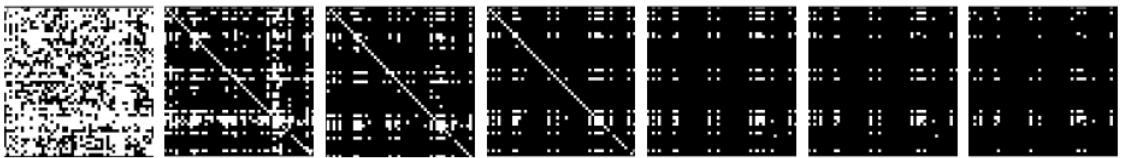
Variational distribution: $q(\mathbf{w}) \sim \mathcal{N}(\boldsymbol{\mu}_q, \sigma_q\mathbf{I})$.

Greedy hyperparameter optimization:

$$\boldsymbol{\mu} = \hat{\mathrm{E}}\mathbf{w}, \quad \sigma = \hat{\mathrm{D}}\mathbf{w}.$$

Parameter pruning w_i using relative PDF:

$$\lambda = \frac{q(\mathbf{0})}{q(\boldsymbol{\mu}_{i,q})} = \exp(-\frac{\mu_i^2}{2\sigma_i^2}).$$



ELBO: normal distribution

"Common" loss function:

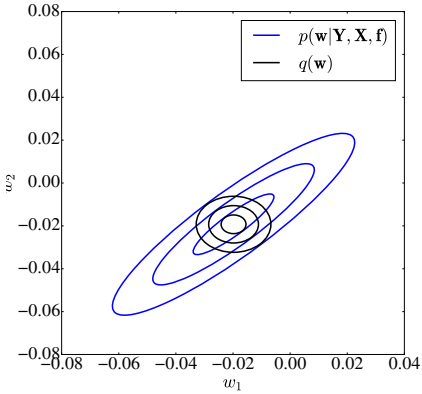
$$L = \sum_{\mathbf{x},\mathbf{y} \in \mathfrak{D}} -\log p(\mathbf{y}|\mathbf{x},\mathbf{w}) + \lambda ||\mathbf{w}||_2^2.$$

Variational inference with $(p(\mathbf{w}|\mathbf{h}) \sim \mathcal{N}(\mathbf{0},\mathbf{1}))$:

$$L = \sum_{\mathbf{x},\mathbf{y}} \log\, p(\mathbf{y}|\mathbf{x},\hat{\mathbf{w}}) +$$

$$+ \frac{1}{2}\left(\mathrm{tr}(\mathbf{A}_q) + \boldsymbol{\mu}_q^{\mathsf{T}} \mathbf{A}^{-1} \boldsymbol{\mu}_q - \ln\, |\mathbf{A}_q|\right).$$

Poor approximation example q



Local reparametrization

How to calculate $\mathrm{E}_q \log p(\mathbf{y}|\mathbf{X}, \mathbf{w})$?

- Graves, 2011: 1 sample per iteration. Use the following properties:

$$w \sim \mathcal{N}(\mu, \sigma^2) \to w \sim \varepsilon\sigma + \mu, \quad \varepsilon \sim \mathcal{N}(0, 1).$$

  - Poor expectation approximation
- Naive solution: sample 1 iteration per element in batch
  - BackProp will be very slow

Local reparametrization, Kingma et al., 2015

Let $y = \text{ReLU}(\mathbf{XW})$ and parameter matrix $\mathbf{W}$ be distributed normally: $w_{i,j} \sim \mathcal{N}(\mu_{i,j}, \sigma^2_{i,j})$.
Then $\mathbf{XW}$ is a Gaussian matrix:

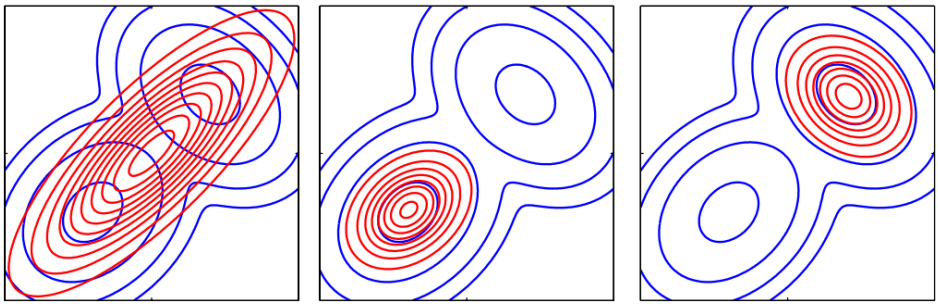
$$\mathbf{G} = \mathbf{XW}, \quad G_{i,j} \sim \mathcal{N}(\sum_k x_{i,k}\mu_{k,j}, \sum_k x^2_{i,k}\sigma^2_{k,j}).$$

Instead of sampling parameters, sample elements from $\mathbf{G}$ (units after activation).

Example

Batch size $= 64$, matrix $\mathbf{W}$ dim is 64×64.

- Graves: one sample, $64 \times 64 = 4096$ elemets. Poor approximation.
- Naive solution: sample parameters 64 times, $64 \times 64 \times 64 = 262144$ elements. Better approximation (in theory).
- Local reparametrization: sample $\mathbf{G}$, $64 \times 64 = 4096$ elements. Better approximation.



Expectation propagation

Minka, 2001: represent prior and approximation distribution via multiplication of factors:

$$p(\mathbf{w}|\mathfrak{D}) = \prod_i f_i, \quad q(\mathbf{w}) = \prod_i \tilde{f}_i$$

Main idea — minimize $KL(p(\mathbf{w}|\mathfrak{D})|q(\mathbf{w}))$.

- Select factor $\tilde{f}_i$ to approximate, «removing» it from consideration, changing into real factor value:
$$q^i \propto f_i \prod_{j \neq i} \tilde{f}_j$$
- Set moments of q^i equal to monents of distribution to approximate (correct, if q is from exponential distribution)
- Repeat until convergence

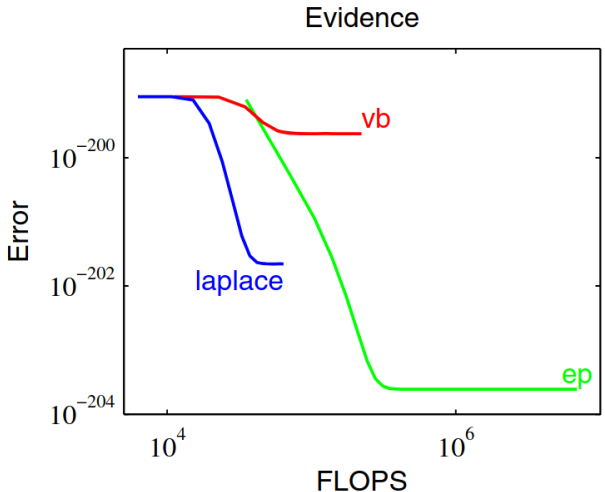
Expectation propagation: pros and cons

Cons:

- Assumption about posterior distribution (rather slight)
- Original version works only for q from exponential distribution
- No convergencce guarantee

Pros:

- Minimizes KL, not it's lower bound



Plot for the 2-component Gaussian mixture.

Probabilistic backpropagation

Combination of Expectiation propagation and backpropagation.[1]

Probabilistic model

Given data $\mathcal{D} = \{x_n, y_n\}_{n=1}^N$, made up of D-dimensional feature vectors and corresponding scalar target variables, we assume that $y_n = f(x_n; W) + \varepsilon_n$, where $f(;\mathbf{W})$ is the output of a multi-layer neural network with weights given by $\mathbf{W}$ and $\varepsilon_n \sim \mathcal{N}(0, \gamma^{-1})$.
Prior distributions:

$$p(W|\lambda) = \prod_{\mathbf{w}\in\mathbf{W}} \mathcal{N}(w|0, \lambda^{-1}),$$

$$p(\lambda) = \Gamma(\lambda|\alpha_0^\lambda, \beta_0^\lambda),$$

$$p(\gamma) = \Gamma(\gamma|\alpha_0^\gamma, \beta_0^\gamma).$$

[1] See talk of Polina Barabanshchikova, 2022

Theory

Likelihood for the weights $\mathbf{W}$ and the noise precision γ is

$$p(\mathbf{y}|\mathbf{W}, \mathbf{X}, \gamma) = \prod_{n=1}^{N} \mathcal{N}(y_n|f(x_n; \mathbf{W}), \gamma^{-1}).$$

The posterior distribution for $\mathbf{W}$, γ, λ

$$p(\mathbf{W}, \gamma, \lambda|\mathcal{D}) = \frac{p(\mathbf{y}|\mathbf{W}, \mathbf{X}, \gamma)p(\mathbf{W}|\lambda)p(\lambda)p(\gamma)}{p(\mathbf{y}|\mathbf{X})}.$$

Probabilistic backpropagation (PBP) approximates the exact posterior with a factored distribution given by

$$q(\mathbf{W}, \gamma, \lambda) = \prod_{\mathbf{w}\in\mathbf{W}} \mathcal{N}(\mathbf{w}|m_w, v_w) \times \Gamma(\gamma|\alpha^\gamma, \beta^\gamma)\Gamma(\lambda|\alpha^\lambda, \beta^\lambda).$$

Method description

Stages of PBP

1. In the first phase, the input data is propagated forward through the network. PBP sequentially approximates the marginal posterior distributions of each weight with a collection of one-dimensional Gaussians that match their marginal means and variances. At the end of this phase, PBP computes the logarithm of the marginal probability of the target variable.

Method description

Stages of PBP

1. In the first phase, the input data is propagated forward through the network. PBP sequentially approximates the marginal posterior distributions of each weight with a collection of one-dimensional Gaussians that match their marginal means and variances. At the end of this phase, PBP computes the logarithm of the marginal probability of the target variable.

2. In the second phase, the gradients of this quantity with respect to the means and variances of the approximate Gaussian posterior are propagated back. These derivatives are used to update the means and variances of the posterior approximation.

Method description

Update rule

Let $f(w)$ encode an arbitrary likelihood function for the single weight w and let our current beliefs regarding the scalar w be captured by a distribution $q(w)$. After seeing the data, our beliefs about w are updated according to Bayes' rule:

$$s(w) = Z^{-1} f(w) q(w),$$

where Z is the normalization constant.
We approximate this posterior with a distribution q^{new} that has the same form as q. The parameters of q^{new} are chosen to minimize the KL divergence between s and q^{new}.

Method description

Update rule (Example)

Assume that $q(w) = \mathcal{N}(w|m, v)$. In this case, the parameters of the new Gaussian beliefs $q^{new}(w) = \mathcal{N}(w|m^{new}, v^{new})$ that minimize the KL divergence between s and q^{new} can be obtained by

$$m^{new} = m + v\frac{\partial \log Z}{\partial m},$$

$$v^{new} = v - v^2\left[\left(\frac{\partial \log Z}{\partial m}\right)^2 - 2\frac{\partial \log Z}{\partial v}\right].$$

Method description

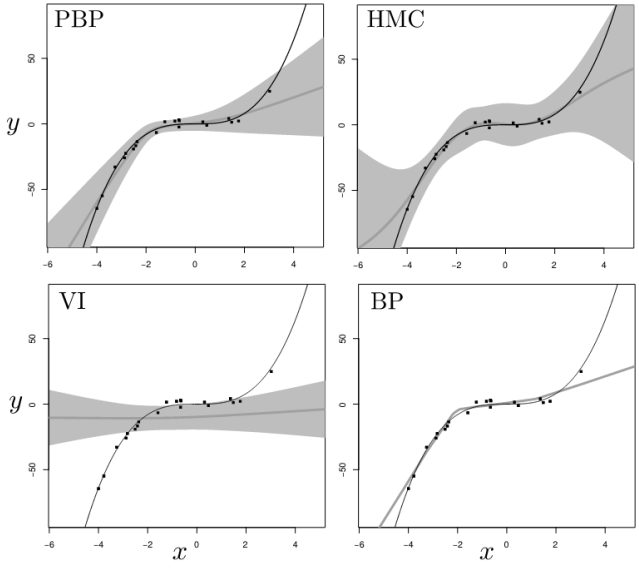
Update rule (Example)

Assume that $q(w) = \mathcal{N}(w|m, v)$. In this case, the parameters of the new Gaussian beliefs $q^{new}(w) = \mathcal{N}(w|m^{new}, v^{new})$ that minimize the KL divergence between s and q^{new} can be obtained by

$$m^{new} = m + v\frac{\partial \log Z}{\partial m},$$

$$v^{new} = v - v^2\left[\left(\frac{\partial \log Z}{\partial m}\right)^2 - 2\frac{\partial \log Z}{\partial v}\right].$$

Remark: Z is approximated during forward pass. Then its derivative is used to update the parameters of marginal distributions.


PBP
HMC
VI
BP
y
x
50
0
−50
−6
−4
−2
0
2
4

Reference

- Bishop C. M. Pattern recognition //Machine learning. – 2006. – Т. 128. – №. 9.
- MacKay D. J. C., Mac Kay D. J. C. Information theory, inference and learning algorithms. – Cambridge university press, 2003.
- Бахтеев О. Ю., Стрижов В. В. Выбор моделей глубокого обучения субоптимальной сложности //Автоматика и телемеханика. – 2018. – №. 8. – С. 129-147.
- Graves A. Practical variational inference for neural networks //Advances in neural information processing systems. – 2011. – Т. 24.
- Louizos C., Ullrich K., Welling M. Bayesian compression for deep learning //arXiv preprint arXiv:1705.08665. – 2017.
- Kingma D. P., Salimans T., Welling M. Variational dropout and the local reparameterization trick //Advances in neural information processing systems. – 2015. – Т. 28. – С. 2575-2583.
- Higgins I. et al. beta-vae: Learning basic visual concepts with a constrained variational framework. – 2016.
- Alemi A. et al. Fixing a broken ELBO //International Conference on Machine Learning. – PMLR, 2018. – С. 159-168.
- Minka T. P. Expectation propagation for approximate Bayesian inference //arXiv preprint arXiv:1301.2294. – 2013.
- Hernández-Lobato J. M., Adams R. Probabilistic backpropagation for scalable learning of bayesian neural networks //International conference on machine learning. – PMLR, 2015. – С. 1861-1869.