Bayesian multimodeling: Variational inference-2

MIPT

2023

Model selection: coherent Bayesian inference

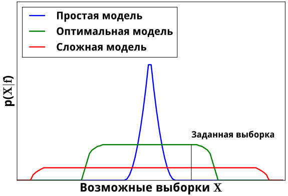
First level: find optimal parameters:

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

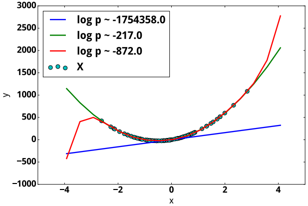
Second level: find optimal model:

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

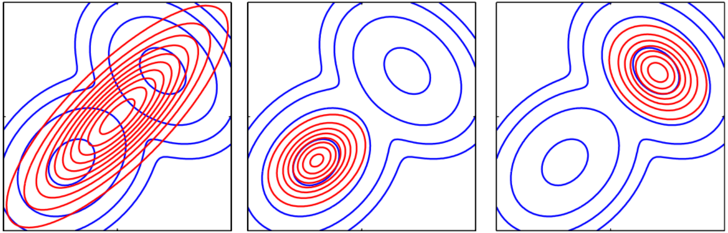
Evidence lower bound, ELBO

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

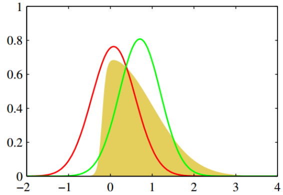
Evidence lower bound estimation often reduces to optimization problem

$$\log \ p(\mathfrak{D}|\mathbf{h}) \geq \mathrm{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathfrak{D})) =$$

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



Variational inference vs. expectation propogation (Bishop)



Laplace Approximation vs Variational inference

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 posteriod distribution $p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \mathbf{h})$:

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

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

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

Outline

- Can we use something except Gaussian distribution?
  - Yes, we can
- Does it need to have an analytical form?
  - No
- Does it need to have some specific properties except continuity?
  - No
- Do we need to optimize KL-divergence w.r.t. distribution parameters for ELBO estimation?
  - No
- Do we need to optimize ELBO for posterior approximation?
  - In general, no

Reparametrization trick

Reparamterization idea:

$$\varepsilon = S_{\theta}(\mathbf{w}), \quad \mathbf{w} = S_{\theta}^{-1}(\varepsilon).$$

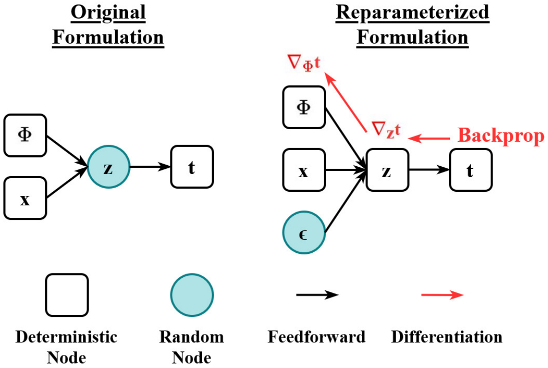
Then:

$$\nabla_{\theta} E_q f(\mathbf{w}) = \mathrm{E}_q \nabla_{\theta} f(S_{\theta}^{-1}(\varepsilon)).$$

Example:

$$w \sim \mathcal{N}(\mu, \sigma^2) \to S(w) = \frac{w - \mu}{\sigma} \sim \mathcal{N}(0, 1).$$

Challenge: calculation of S^{-1} is an expensive operation.



Source: wikipedia

Normalizing Flows

Given an invertible smooth mapping $\mathbf{g}$ (flow) and a distribution $\mathbf{z} \sim q$.
Then $q(\mathbf{g}(\mathbf{z}))$ is a distribution:

$$\mathbf{g}(g(\mathbf{z})) = q(\mathbf{z}) \left(\det \frac{\partial g}{\partial \mathbf{z}} \right)^{-1}.$$

Example: planar flow:

$$\mathbf{g}(\mathbf{z}) = \mathbf{z} + \mathbf{w}_1 \boldsymbol{\sigma}(\mathbf{w}_2{}^{\mathsf{T}} \mathbf{x}).$$

Reparametrization trick

Reparamterization idea:

$$\varepsilon = S_{\theta}(\mathbf{w}), \quad \mathbf{w} = S_{\theta}^{-1}(\varepsilon).$$

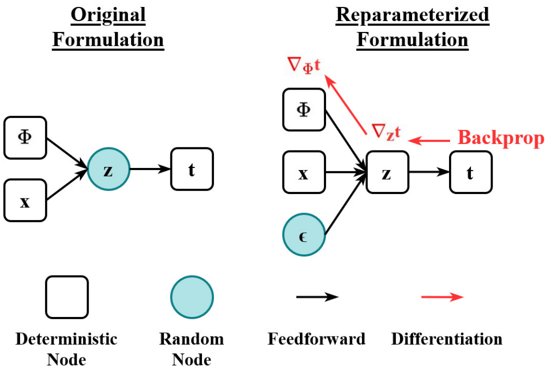
Then:

$$\nabla_{\theta} E_q f(\mathbf{w}) = \mathrm{E}_q \nabla_{\theta} f(S_{\theta}^{-1}(\varepsilon)).$$

Example:

$$w \sim \mathcal{N}(\mu, \sigma^2) \to S(w) = \frac{w - \mu}{\sigma} \sim \mathcal{N}(0, 1).$$

Challenge: calculation of S^{-1} is an expensive operation.



Source: wikipedia

Implicit reparametrization trick

$$\nabla_{\boldsymbol{\theta}} E_q f(\mathbf{w}) = \mathsf{E}_q \nabla_{\mathbf{w}} f(\mathbf{w}) \nabla_{\boldsymbol{\theta}} \mathbf{w}.$$

Use a total gradient formula for $\varepsilon = S_{\boldsymbol{\theta}}(\mathbf{w})$:

$$\nabla_{\mathbf{w}} S_{\boldsymbol{\theta}}(\mathbf{w}) \nabla_{\boldsymbol{\theta}} \mathbf{w} + \nabla_{\boldsymbol{\theta}} S_{\boldsymbol{\theta}}(\mathbf{w}) = 0 \rightarrow$$

$$\rightarrow \nabla_{\boldsymbol{\theta}} \mathbf{w} = -(\nabla_{\mathbf{w}} S_{\boldsymbol{\theta}}(\mathbf{w}))^{-1} \nabla_{\boldsymbol{\theta}} S_{\boldsymbol{\theta}}.$$

Obtain an expression without inverse function for S.
For 1d samples we can use, for example:

$$S(\mathbf{w}) = F(\mathbf{w}|\boldsymbol{\theta}) \sim \mathcal{U}(0, 1).$$

Table 4: Test negative log-likelihood (lower is better) for VAE on MNIST. Mean $\pm$ standard deviation over 5 runs. The von Mises-Fisher results are from [9].

Prior	Variational posterior	$D = 2$	$D = 5$	$D = 10$	$D = 20$	$D = 40$
$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\sigma^2)$	131.1 ± 0.6	107.9 ± 0.4	92.5 ± 0.2	88.1 ± 0.2	88.1 ± 0.0
$\mathrm{Gamma}(0.3, 0.3)$	$\mathrm{Gamma}(\alpha,\beta)$	132.4 ± 0.3	108.0 ± 0.3	94.0 ± 0.3	90.3 ± 0.2	90.6 ± 0.2
$\mathrm{Gamma}(10, 10)$	$\mathrm{Gamma}(\alpha,\beta)$	135.0 ± 0.2	107.0 ± 0.2	92.3 ± 0.2	88.3 ± 0.2	88.3 ± 0.1
$\mathrm{Uniform}(0, 1)$	$\mathrm{Beta}(\alpha,\beta)$	128.3 ± 0.2	107.4 ± 0.2	94.1 ± 0.1	88.9 ± 0.1	88.6 ± 0.1
$\mathrm{Beta}(10, 10)$	$\mathrm{Beta}(\alpha,\beta)$	131.1 ± 0.4	$\mathbf{106.7} \pm 0.1$	$\mathbf{92.1} \pm 0.2$	$\mathbf{87.8} \pm 0.1$	$\mathbf{87.7} \pm 0.1$
$\mathrm{Uniform}(-\pi,\pi)$	$\mathrm{vonMises}(\mu,\kappa)$	$\mathbf{127.6} \pm 0.4$	107.5 ± 0.4	94.4 ± 0.5	90.9 ± 0.1	91.5 ± 0.4
$\mathrm{vonMises}(0, 10)$	$\mathrm{vonMises}(\mu,\kappa)$	130.7 ± 0.8	107.5 ± 0.5	92.3 ± 0.2	$\mathbf{87.8} \pm 0.2$	87.9 ± 0.3
$\mathrm{Uniform}(S^D)$	$\mathrm{vonMisesFisher}(\boldsymbol{\mu},\kappa)$	132.5 ± 0.7	108.4 ± 0.1	93.2 ± 0.1	89.0 ± 0.3	90.9 ± 0.3

MCMC and variational inference

MCMC idea: Sample from the simple distribution and accpet them, if the ratio is greater than some threshold:

$$\min\left(1, \frac{p(\mathbf{w}^\tau|\mathbf{y},\mathbf{X},\mathbf{h})}{p(\mathbf{w}^{\tau-1}|\mathbf{y},\mathbf{X},\mathbf{h})}\right),$$

where $\mathbf{w}^\tau$ is set based on the previous sample:

$$\mathbf{w}^\tau = T(\mathbf{w}^{\tau-1}).$$

Salimans et al., 2014: let's interperete the sequence of some operator T application as a variational optimization:

$$T^1 \circ \dots T^\eta(\mathbf{w}) \to p(\mathbf{w}^\tau|\mathbf{y},\mathbf{X},\mathbf{h}).$$

Maclaurin et. al, 2015: use gradient descent as such operator. Do not reject samples at all.

Optimization operator, Maclaurin et. al, 2015

Definition

Let T be an algorithm of changing model parameters $\mathbf{w}'$ using previous parameter values $\mathbf{w}$:

$$\mathbf{w}' = T(\mathbf{w}).$$

Definition

Let L be a continuos loss function.
Define a gradient descent operator in the following way:

$$T(\mathbf{w}) = \mathbf{w} - \beta \nabla L(\mathbf{w}, \mathbf{y}, \mathfrak{D}).$$

Gradient descent for evidence estimation

Consider posterior probability maximization:

$$L = -\log\, p(\mathfrak{D}, \mathbf{w}|\mathbf{h}) = -\sum_{\mathfrak{D} \in \mathfrak{D}} \log\, p(\mathfrak{D}|\mathbf{w}, \mathbf{h}) p(\mathbf{w}|\mathbf{h})$$

Optimize neural network in a multi-start regime with r initial parameter values $\mathbf{w}_1, \dots, \mathbf{w}_r$ using (stochastic) gradient descent:

$$\mathbf{w}' = T(\mathbf{w}).$$

The parameter vectors $\mathbf{w}_1, \dots, \mathbf{w}_r$ are from some latent distribution $q(\mathbf{w})$.

Entropy

We can rewrite variational inference using differential entropy term:

$$\log p(\mathfrak{D}|\mathbf{f}) \geq \int_{\mathbf{w}} q(\mathbf{w}) \log \frac{p(\mathfrak{D}, \mathbf{w}|\mathbf{h})}{q(\mathbf{w})} d\mathbf{w} =$$

$$\mathsf{E}_{q(\mathbf{w})}[\log p(\mathfrak{D}, \mathbf{w}|\mathbf{h})] + \mathsf{S}(q(\mathbf{w})),$$

where $\mathsf{S}(q(\mathbf{w}))$ is a differential entropy:

$$\mathsf{S}(q(\mathbf{w})) = -\int_{\mathbf{w}} q(\mathbf{w}) \log q(\mathbf{w}) d\mathbf{w}.$$

Gradient descent for evidence estimation

Statement

Let L be a Lipschitz function, and optimization operator be a bijection. Then entropy difference for two steps is:

$$\mathsf{S}(q'(\mathbf{w})) - \mathsf{S}(q(\mathbf{w})) \simeq \frac{1}{r}\sum_{g=1}^{r}\bigl(-\beta \mathrm{Tr}[\mathbf{H}(\mathbf{w}'^{g})] - \beta^2 \mathrm{Tr}[\mathbf{H}(\mathbf{w}'^{g})\mathbf{H}(\mathbf{w}'^{g})]\bigr).$$

Final estimation for the τ optimization step:

$$\log \hat{p}(\mathbf{Y}|\mathfrak{D}, \mathbf{h}) \sim \frac{1}{r}\sum_{g=1}^{r} L(\mathbf{w}_{\tau}^{g}, \mathfrak{D}, \mathbf{Y}) + \mathsf{S}(q^0(\mathbf{w})) +$$

$$+\frac{1}{r}\sum_{b=1}^{\tau}\sum_{g=1}^{r}\bigl(-\beta \mathrm{Tr}[\mathbf{H}(\mathbf{w}_b^{g})] - \beta^2 \mathrm{Tr}[\mathbf{H}(\mathbf{w}_b^{g})\mathbf{H}(\mathbf{w}_b^{g})]\bigr),$$

$\mathbf{w}_b^g$ is a parameter vector for optimization g on the step b, $\mathsf{S}(q^0(\mathbf{w}))$ is an initial entropy.

How to calculate Hessian trace?

Problem

$$\mathrm{Tr}[\mathbf{H}(\mathbf{w}_b^g)]$$

Statement

Let $\mathbf{U}$ be a symmetric matrix and $\mathbf{v}$ be the random vector with the following properties:

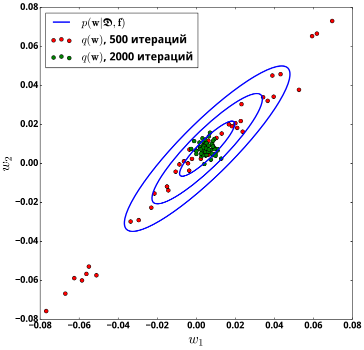
1. $\mathrm{E}v_i = 0$;
2. $Var(v_i) = 1$.

Then

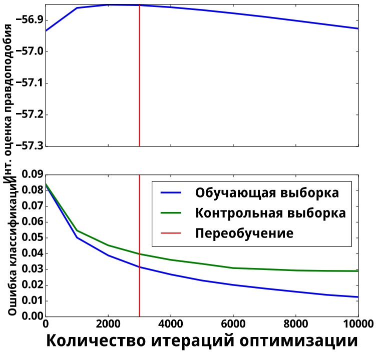
$$\mathrm{E}\mathbf{v}^{\mathrm{T}}\mathbf{U}\mathbf{v} = Tr[\mathbf{U}].$$

Overfitting, Maclaurin et. al, 2015

Gradient descent does not optimize KL-divergence $\text{KL}(q(\mathbf{w})||p(\mathbf{w}|\mathfrak{D},\mathbf{h}))$. Evidence estimation gets worse while optimization tends to the optimal parameter values. This can be considered as a overfitting start.



Convergence



Overfitting start

Stochastic gradient Langevin dynamics

A modification of SGD:

$$T = \mathbf{w} - \beta \nabla L + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \frac{\beta}{2})$$

where β changes with a number of iterations:

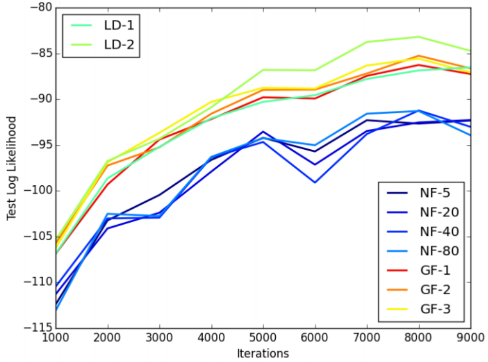
$$\sum_{\tau=1}^{\infty} \beta_\tau = \infty, \quad \sum_{\tau=1}^{\infty} \beta_\tau^2 < \infty.$$

Statement [Welling, 2011]. Distribution $q^\tau(\mathbf{w})$ converges to posterior distribution $p(\mathbf{w}|\mathbf{X}, \mathbf{f})$.
Entropy adjustment:

$$\hat{\mathsf{S}}(q^\tau(\mathbf{w})) \geq \frac{1}{2}|\mathbf{w}|\log(\exp(\frac{2\mathsf{S}(q^\tau(\mathbf{w}))}{|\mathbf{w}|}) + \exp(\frac{2\mathsf{S}(\epsilon)}{|\mathbf{w}|})).$$

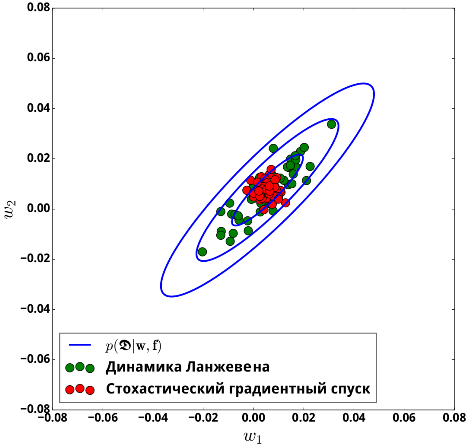
Stochastic gradient Langevin dynamics for generative models

Altieri et al., 2015: sample latent variable $\mathbf{z}$ and use SGLD as a normalizing flow.



SGLD vs SGD

Parameter distribution after 2000 iterations:



Stein operator

Given a smooth probability function p and a smooth vector function $\boldsymbol{\phi}$. Define a Stein operator as the following:

$$\mathcal{A}_p\boldsymbol{\phi}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})\boldsymbol{\phi}^{\mathsf{T}} + \nabla_{\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{x}).$$

Stein's identity:

$$\mathsf{E}_{\mathbf{x}\sim p}\mathcal{A}_p\boldsymbol{\phi}(\mathbf{x}) = 0.$$

If we use q instead of p in the $\mathcal{A}_p$ we get a non-zero result, but close to zero as soon as p is close to q.
Let $T(\mathbf{x}) = \mathbf{x} + \varepsilon\boldsymbol{\phi}(\mathbf{x})$. Then:

$$\nabla_\varepsilon KL(q||p)|_{\varepsilon=0} = \mathsf{E}_{\mathbf{x}\sim q}\mathrm{trace}\mathcal{A}_p\boldsymbol{\phi}.$$

Given a kernel $\mathbf{K}$, the optimal ϕ for minimizing KL is:

$$\phi^*(\mathbf{x}') = \mathsf{E}_{\mathbf{x}\sim q}\nabla_{\mathbf{x}} \log\ p(\mathbf{x})\mathbf{K}(\mathbf{x},\mathbf{x}') + \nabla_{\mathbf{x}}\mathbf{K}(\mathbf{x},\mathbf{x}').$$

Stein operator: algorithm

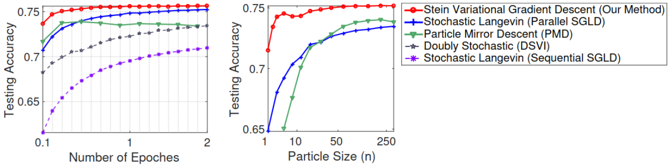
Algorithm 1 Bayesian Inference via Variational Gradient Descent

Input: A target distribution with density function $p(x)$ and a set of initial particles $\{x_i^0\}_{i=1}^n$.
Output: A set of particles $\{x_i\}_{i=1}^n$ that approximates the target distribution $p(x)$.
for iteration ℓ **do**

$$x_i^{\ell+1} \leftarrow x_i^\ell + \epsilon_\ell \hat{\phi}^*(x_i^\ell) \quad \text{where} \quad \hat{\phi}^*(x) = \frac{1}{n}\sum_{j=1}^n \left[k(x_j^\ell, x)\nabla_{x_j^\ell} \log p(x_j^\ell) + \nabla_{x_j^\ell} k(x_j^\ell, x) \right], \quad (8)$$

where ϵ_ℓ is the step size at the ℓ-th iteration.
end for

Stein operator: results



(a) Particle size $n = 100$ (b) Results at 3000 iteration ($\approx$ 0.32 epoches)

	Avg. Test RMSE		Avg. Test LL		Avg. Time (Secs)	
Dataset	**PBP**	**Our Method**	**PBP**	**Our Method**	**PBP**	**Ours**
Boston	2.977 ± 0.093	$\mathbf{2.957 \pm 0.099}$	-2.579 ± 0.052	$\mathbf{-2.504 \pm 0.029}$	18	**16**
Concrete	5.506 ± 0.103	$\mathbf{5.324 \pm 0.104}$	-3.137 ± 0.021	$\mathbf{-3.082 \pm 0.018}$	33	**24**
Energy	1.734 ± 0.051	$\mathbf{1.374 \pm 0.045}$	-1.981 ± 0.028	$\mathbf{-1.767 \pm 0.024}$	25	**21**
Kin8nm	0.098 ± 0.001	$\mathbf{0.090 \pm 0.001}$	0.901 ± 0.010	$\mathbf{0.984 \pm 0.008}$	118	**41**
Naval	0.006 ± 0.000	$\mathbf{0.004 \pm 0.000}$	3.735 ± 0.004	$\mathbf{4.089 \pm 0.012}$	173	**49**
Combined	4.052 ± 0.031	$\mathbf{4.033 \pm 0.033}$	-2.819 ± 0.008	$\mathbf{-2.815 \pm 0.008}$	136	**51**
Protein	4.623 ± 0.009	$\mathbf{4.606 \pm 0.013}$	-2.950 ± 0.002	$\mathbf{-2.947 \pm 0.003}$	682	**68**
Wine	0.614 ± 0.008	$\mathbf{0.609 \pm 0.010}$	-0.931 ± 0.014	$\mathbf{-0.925 \pm 0.014}$	26	**22**
Yacht	$\mathbf{0.778 \pm 0.042}$	0.864 ± 0.052	$\mathbf{-1.211 \pm 0.044}$	-1.225 ± 0.042	25	25
Year	$8.733 \pm \text{NA}$	$\mathbf{8.684 \pm NA}$	$-3.586 \pm \text{NA}$	$\mathbf{-3.580 \pm NA}$	7777	**684**

References

- Bishop C. M. Pattern recognition //Machine learning. – 2006. – T. 128. – №. 9.
- MacKay D. J. C., Mac Kay D. J. C. Information theory, inference and learning algorithms. – Cambridge university press, 2003.
- Salimans, Tim, Diederik Kingma, and Max Welling, 2015. Markov chain monte carlo and variational inference: Bridging the gap
- Altieri: http://approximateinference.org/accepted/AltieriDuvenaud2015.pdf
- Stephan Mandt, Matthew D. Hoffman, David M. Blei, 2017. Stochastic Gradient Descent as Approximate Bayesian Inference
- Бахтеев О. Ю., Стрижов В. В. Выбор моделей глубокого обучения субоптимальной сложности //Автоматика и телемеханика. – 2018. – №. 8. – С. 129-147.
- Figurnov M., Mohamed S., Mnih A. Implicit reparameterization gradients //arXiv preprint arXiv:1805.08498. – 2018.
- Jang E., Gu S., Poole B. Categorical reparameterization with gumbel-softmax //arXiv preprint arXiv:1611.01144. – 2016.
- Potapczynski A., Loaiza-Ganem G., Cunningham J. P. Invertible gaussian reparameterization: Revisiting the gumbel-softmax //arXiv preprint arXiv:1912.09588. – 2019.
- Maddison C. J., Mnih A., Teh Y. W. The concrete distribution: A continuous relaxation of discrete random variables //arXiv preprint arXiv:1611.00712. – 2016.
- Shayer O., Levi D., Fetaya E. Learning discrete weights using the local reparameterization trick //arXiv preprint arXiv:1710.07739. – 2017.
- Li Y., Turner R. E. Rényi Divergence Variational Inference //arXiv preprint arXiv:1602.02311. – 2016.
- Hutchinson, M. F. (1990). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Communications in Statistics - Simulation and Computation, 19(2), 433–450.
- Rezende, D., Mohamed, S. (2015, June). Variational inference with normalizing flows. In International conference on machine learning (pp. 1530-1538). PMLR.
- Liu, Q., Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. Advances in neural information processing systems, 29.