Lomonosov Moscow State University
Faculty of Computer Science

Review of matherials on

Gaussian Processes for Machine Learning

Pavel Izmailov

Moscow, 2016

1 Theory

In this section an introduction to Gaussian process theory is provided.

1.1 Gaussian Process

Consider the following definition

Definition 1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

A Gaussian process is completely specified by it's mean function and covariance function. These functions are defined as follows

Definition 2. *Let $f(x)$ be a real-valued Gaussian process. Then the functions*

$$m(x) = \mathbb{E}[f(x)],$$
$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))],$$

are the mean function and the covariance function of the process f respectively.

We will write the Gaussian process as $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$.

1.2 GP-regression

Consider the following task. We have a dataset $\{(x_i, f_i)|i = 1, \ldots, n\}$, generated from a Gaussian process $f \sim \mathcal{GP}(m(x), k(x, x'))$, let $x \in \mathbb{R}^d$. We will denote the matrix comprised of points $x_1, \ldots, x_n$ by $X \in \mathbb{R}^{n \times d}$ and the vector of corresponding values $f_1, ..., f_n$ by $f \in \mathbb{R}^n$. We want to predict the values $f_* \in \mathbb{R}^m$ of this random process at a set of other m points $X_* \in \mathbb{R}^{m \times d}$. The joint distribution of f and f_* is given by

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right),$$

where $K(X, X) \in \mathbb{R}^{n \times n}$, $K(X, X_*) = K(X^*, X)^T \in \mathbb{R}^{n \times m}$, $K(X^*, X^*) \in \mathbb{R}^{m \times m}$ are the matrices comprised of pairwise values of the covariance function k for the given sets.

The conditional distribution

$$f_*|X_*, X, f \sim \mathcal{N}(\hat{m}, \hat{K}),$$

where

$$\mathbb{E}[f_*|f] = \hat{m} = K(X_*, X)K(X, X)^{-1}f,$$
$$\mathrm{cov}(f_*|f) = \hat{K} = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*).$$

Thus, predicting the values of the Gaussian process at a new data point requires solving a linear system with a matrix of size $n \times n$ and thus scales as $O(n^3)$.

In fig. 1 you can see the examples of one and two-dimensional gaussian-processes, reconstructed from the data. The data points are shown by black '+' signs.

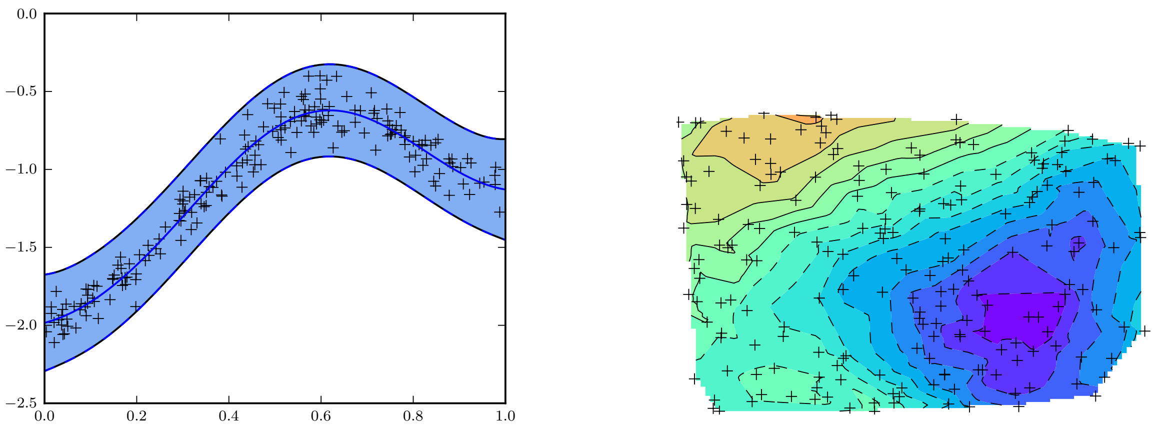

Рис. 1: One and two-dimensional gaussian processes

1.2.1 Noisy case

Consider the following model. We now have a dataset $\{(x_i, y_i)|i = 1, \dots n\}$, where $y_i = f(x_i) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_n)$. This means that we only have access to the noisy observations and not the true values of the process at data points. With the notation and logics similar to the one we used it the previous section we can find the conditional distribution for the values f_* of the process at new points X_* in this case:

$$f_*|y \sim \mathcal{N}(\hat{m}, \hat{K}),$$

$$\mathbb{E}[f_*|y] = \hat{m} = K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} y,$$
$$\operatorname{cov}(f_*|y) = \hat{K} = K(X_*, X_*) - K(X_*, X)(K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*).$$

1.3 GP-classification

Another important class of problems in machine learning is classification. We will consider the following problem. We have a dataset $\{(x_i, y_i)|i = 1, \dots, n\}$, where $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$. We want to predict the probabilities of new datapoints x_* belonging to positive class.

We will consider the following model. We will introduce a latent function $f(x)$ and put a GP prior over it. The model is then

$$\pi(x) = p(y_* = +1|x_*) = \sigma(f(x_*)),$$

where $f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$, and $\sigma(z) = (1 + \exp(-z))^{-1}$ (one can use other sigmoid functions as well).

Now inference can be done in two steps. First, we should find the conditional distribution of the value of the latent process f at the new data point x_*. This can be computed as follows

$$p(f_*|X, y, x_*) = \int p(f_*|X, x_*, f)p(f|X, y)df. \tag{1}$$

Now, the probability of the positive class is given by marginalizing over the latent variable f_*.

$$\pi(x_*) = p(y = +1|X, y, x_*) = \int \sigma(f_*)p(f_*|X, y, x_*)df_*. \tag{2}$$

Unfortunantely, both the integrals in 1 and 2 are intractable. Thus, we have to use various techniques to approximate these integrals. We will describe a method, based on Laplace approximation below.

1.3.1 Laplace approximation

Laplace approximation for approximating the integral 1 utilizes a Gaussian approximation $q(f|X, y)$ to the posterior $p(f|X, y)$. This approximation is obtained via Taylor expansion of $\log p(f|X, y)$ around it's maximum.

In order to find the maximum of the posterior, we first use the Bayes rule.

$$p(f|X, y) = \frac{p(y|f)p(f|X)}{p(y|X)}.$$

However, the denomenator does not depend on X, so the maximum of the posterior can be found as the maximum of the function

$$\Psi(f) = \log(p(y|f)p(f|X)) = \log p(y|f) + \log p(f|X) = \log p(y|f) - \frac{1}{2}f^T K^{-1} f - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi,$$

where $K = K(X, X)$. Note, that $p(y|f) = \prod_{i=1}^n p(y_i|f_i)$

Now, differentiating Ψ wrt f we obtain

$$\frac{\partial \Psi}{\partial f} = \frac{\partial \log p(y|f)}{\partial f} - K^{-1}f,$$

$$\frac{\partial^2 \Psi}{\partial f^2} = \frac{\partial^2 \log p(y|f)}{\partial f^2} - K^{-1}.$$

For the logistic likelihood $p(y_i|f_i) = -\log(1 + \exp(-y_i f_i))$ we obtain

$$\frac{\partial \log p(y_i|f_i)}{\partial f_i} = \frac{y_i + 1}{2} - p(y_i = +1|f_i),$$

$$\frac{\partial^2 \log p(y|f)}{\partial f^2} = -p(y_i = +1|f_i)(1 - p(y_i = +1|f_i)).$$

Setting the gradient of Ψ to zero we obtain

$$\left.\frac{\partial \Psi(f)}{\partial f}\right|_{f=\hat{f}} = 0 \Rightarrow K\left(\left.\frac{\partial \log p(y_i|f_i)}{\partial f_i}\right|_{f=\hat{f}}\right). \tag{3}$$

This equation is not analytically solvable, but might be useful later. In order to find the optimum of Ψ we use the Newton method.

Having found the maximum $\hat{f}$ of the posterior, we can now specify the Laplace approximation to the posterior as

$$q(f|X,y) = \mathcal{N}\left(\hat{f}, -\left(\left.\frac{\partial^2\Psi(f)}{\partial f^2}\right|_{f=\hat{f}}\right)^{-1}\right).$$

Now, we can estimate the probabilities by $\pi(x_*) = \int \sigma(f_*)q(f_*|X,y,x_*)df$, or just use $\pi(x_*) = \sigma(\hat{f})$, if we are only interested in the most probable classification and not the probabilities themselves (it's shown that for the most probable classifications the two approaches are equivalent).

1.4 Kernel functions

To be wrritten.

1.5 Hyper-parameter estimation

Bayesian paradigm provides a way of estimating the kernel hyper-parameters of the GP-model through maximizization of the marginal likelihood of the model. Marginal likelihood is given by

$$p(y|X) = \int p(y|f,X)p(f|X)df,$$

which is the likelihood, marginalized over the hidden values f of the underlying process.

For the GP-regression the marginal likelihood can be computed in claused form and is given by

$$\log p(y|X) = -\frac{1}{2}y^T(K+\sigma_n^2 I)^{-1}y - \frac{1}{2}\log|K+\sigma_n^2 I| - \frac{n}{2}\log 2\pi. \tag{4}$$

For the method, described in section 1.3 the marginal likelihood can be computed as follows.

$$p(y|X) = \int p(y|f,X)p(f|X)df = \int \exp\Psi(f)df,$$

where we use the notation from section 1.3. Using the Taylor expansion, locally near $\hat{f}$ we have $\Psi(f) \simeq \Psi(\hat{f}) + \frac{1}{2}(f-\hat{f})^T A(f-\hat{f})$, where A is the hessian of Ψ at $\hat{f}$. Using this approximation we obtain

$$p(y|X) \simeq q(y|X) = \exp(\Psi(\hat{f}))\int \exp(-\frac{1}{2}(f-\hat{f})^T A(f-\hat{f}))df.$$

This last integral can be calculated analytically to obtain a closed form approximation to the log marginal likelihood.

$$\log q(y|X) = -\frac{1}{2}\hat{f}^T K^{-1}\hat{f} + \log p(y|\hat{f}) - \frac{1}{2}\log|B|, \tag{5}$$

where

$$|B| = |K| \left| - \frac{\partial^2 \log p(y|f)}{\partial f^2} \bigg|_{f=\hat{f}} \right|.$$

Using the derived formulas 5 and 4 we can find the optimal values of hyper-parameters through maximization of the marginal likelihood of the corresponding model.

1.6 Theoretical perspectives

To be wrritten.

2 Review of existing methods

It follows from the discussion above, that full Gaussian process regression scales as $O(n^3)$ and thus cannot be applied to big datasets. In this section we will review several approximate methods, that make Gaussian processes practical.

2.1 Methods, based on inducing inputs

Most of the existing methods are based on introducing a set of m function points that are called inducing inputs. Using these inputs one can make approximate predictions of the values of the hidden process at test points with a complexity of $O(nm^3)$ instead of $O(n^3)$.

Consider the following situation. We have a dataset of n examples x_i with corresponding values y_i. We will denote the matrix of pairwise values of the covariance function by K_{nn}. Now we introduce a set of m inducing inputs. We will denote the corresponding covariance matrix by K_{mm} and the matrices of covariances between the inducing points and training points by K_{nm} and K_{mn}. We will denote the vectors, comprised of noisy and true function values y_i and f_i at training points by y and f respectively. We will also introduce a distribution $q(u)$ over the hidden function values u at the inducing inputs.

It's easy to see, that

$$p(y|f) = \mathcal{N}(y|f, \sigma_n I),$$
$$p(f|u) = \mathcal{N}(f|K_{nm}K_{mm}^{-1}u, \tilde{K}),$$
$$p(u) = \mathcal{N}(u|0, K_{mm}),$$

where $\tilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$.

2.1.1 Variational learning of inducing points

The method discussed here was introduced in [1]. This method provides a way to find the optimal positions of the inducing points, as well as the optimal distribution of the process value at these points.

Let z denote a vector comprized of the process values at some new points. We can calculate the predictive distribution at these points as follows

$$p(z|y) = \int p(z|f)p(f|y)df.$$

Let's fix the inducing point positions $x_1, \ldots, x_m$. As above, u is the vector compised of the process values at these points. We can rewrite the above equation

$$p(z|y) = \iint p(z|u, f)p(f|u, y)p(u|y)df du, \tag{6}$$

as $p(z|u, f, y) = p(z|u, f)$.

Suppose for a moment, that u is a sufficient statistics for the parameter f in the scence that z and f are conditionally independent given u. Then we have

$$p(z|f,u) = \frac{p(z,f|u)}{p(f|u)} = \frac{p(z|u)p(f|u)}{p(f|u)} = p(z|u),$$

$$p(z|y,u) = \frac{p(z,y,u)}{p(y,u)} = \frac{\int p(y|f)p(f,z,u)du}{\iint p(y|f)p(f,z,u)df dz} = \frac{\int p(y|f)p(z|u)p(u|f)p(f)df}{\iint p(y|f)p(z|u)p(u|f)p(f)df dz} =$$
$$= \frac{\int p(y|f)p(f)p(u|f)df \cdot p(z|u)}{\int p(y|f)p(f)p(u|f)df \cdot \int p(z|u)dz} = \frac{\int p(y,f)p(u|f)df}{\int p(y,f)p(u|f)df} p(z|u) = p(z|u).$$

So, $p(z|y,u) = p(z|u)$. If we set the points, corrwsponding to the process values z, to the traing points, we will have $z = f$, and thus $p(f|y,u) = p(f|u)$.

Substituting these formulas into (6) we achieve

$$q(z) = p(z|y) = \iint p(z|u)p(f|u)p(u|y)df du = \iint p(z|u)p(u|y)du =$$
$$= \int p(z|u)\varphi(u)du = \int q(z,u)du, \tag{7}$$

where $\varphi(u) = p(u|y)$, $q(z,u) = p(z|u)\varphi(u)$.

In practice however it's difficult to guarantee that u is a sufficient statistics. Thus we can only expect $q(z)$ to be an approximation to $p(z|y)$. In such case we can choose $\varphi(u)$ to be a variational distribution, where in general $\varphi(u) \neq p(u|y)$. We will consider $\varphi(u)$ to be Gaussian with a mean vector μ and covariance matrix Σ.

By using the eq. (7) we can calculate the approximate posterior GP mean at point x and covariance at points x, x'

$$\mathbb{E}[z(x)] = K_{xm}K_{mm}^{-1}\mu,$$
$$\mathrm{cov}(z(x), z(x')) = k(x,x') - K_{xm}K_{mm}^{-1}K_{mx'} + K_{xm}AK_{mx'},$$

where $A = K_{mm}^{-1}\Sigma K_{mm}^{-1}$.

Now we have to specify a way to find the variational distribution parameters μ and Σ, and the inducing input positions X_m and a way to optimize the kernel hyper-parameters. In order to do so, we will form the variational distribution $q(f,u)$ and the exact posterior $p(f,u|y)$ on the training function values and the values at the inducing points, and then minimize the KL-divergence between these two distributions. This minimization is equivalently expressed as the maximization of the following lower bound of the true marginal likelihood:

$$F_V(X_m, \varphi) = \iint p(f|u)\varphi(u) \log \frac{p(y|f)p(u)}{\varphi(u)} df du.$$

This bound can be optimized analytically with respect to ϕ. The optimal distribution $\varphi(u) \sim \mathcal{N}(u|\hat{u}, \Lambda^{-1})$, where

$$\Lambda = \frac{1}{\sigma_n} K_{mm}^{-1}K_{mn}K_{nm}K_{mm}^{-1} + K_{mm}^{-1},$$

$$\hat{u} = \frac{1}{\sigma_n}\Lambda^{-1}K_{mm}^{-1}K_{mn}y.$$

Substituting the optimal values of variational parameters into the F_V we obtain the following bound

$$F_V(X_m) = \log \mathcal{N}(y|0, \sigma_n^2 I + K_{nm}K_{mm}^{-1}K_{mn}) - \frac{1}{2\sigma_n^2}\mathrm{tr}(\tilde{K}).$$

Another derivation of this lower bound is provided in section (2.1.2).

The bound $F_V(X_m)$ is computed in $o(nm^2)$ time. Now we will calculate it's gradient in order to be able to maximize it with respect to X_m and kernel hyper-parameters. We will denote $B = \sigma_n^2 I + K_{nm}K_{mm}^{-1}K_{mn}$. Then

$$F_V(X_m, \theta, \sigma_n) = -\frac{1}{2}\left(n \log 2\pi + \log|B| + y^T B^{-1} y + \frac{1}{\sigma_n^2}\mathrm{tr}(\tilde{K})\right),$$

$$\frac{\partial F_V}{\partial \theta} = \frac{1}{2}\left(-\mathrm{tr}\left(B^{-1}\frac{\partial B}{\partial \theta}\right) + y^T B^{-1}\frac{\partial B}{\partial \theta}B^{-1}y - \right.$$
$$\left. - \frac{1}{\sigma_n^2}\mathrm{tr}\left(\frac{\partial K_{nn}}{\partial \theta} - \left(\frac{\partial K_{nm}}{\partial \theta}K_{mm}^{-1} - K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial \theta}K_{mm}^{-1}\right)K_{mn} - K_{nm}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial \theta}\right)\right),$$

where

$$\frac{\partial B}{\partial \theta} = \left(\frac{\partial K_{nm}}{\partial \theta}K_{mm}^{-1} - K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial \theta}K_{mm}^{-1}\right)K_{mn} + K_{nm}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial \theta}.$$

We can rewrite

$$\frac{\partial F_V}{\partial \theta} = \frac{1}{2}\left(-\mathrm{tr}\left(B^{-1}\frac{\partial B}{\partial \theta}\right) + y^T B^{-1}\frac{\partial B}{\partial \theta}B^{-1}y - \frac{1}{\sigma_n^2}\mathrm{tr}\left(\frac{\partial K_{nn}}{\partial \theta} - \frac{\partial B}{\partial \theta}\right)\right).$$

Now we can optimize F_V with respect to kernel hyper-parameters. Similarly, we can take derivatives with respect to X_m and σ_n and opptimize F_V with respect to them as well.

However, if we compute F_v and it's derivatives as they are, it takes $O(n^3)$ time which is not faster, than recovering the full Gaussian process. So, we have to rewrite these values in a form that allows for faster computation.

First of all, let's deal with $\log|B|$ and B^{-1}. Using the matrix determinant lemma we obtain

$$|B| = |\sigma_n^2 I + K_{nm}K_{mm}^{-1}K_{mn}| = \frac{\left|K_{mm} + \frac{K_{mn}K_{nm}}{\sigma_n^2}\right|\sigma_n^2}{|K_{mm}|}.$$

So, denoting $A = K_{mm} + \frac{K_{mn}K_{nm}}{\sigma_n^2}$, we obtain

$$\log|B| = \log|A| + 2\log\sigma_n - \log|K_{mm}|.$$

Note tha this is computed in $O(nm^2)$ instead of $O(n^3)$.

Using the Woodbury identity, we obtain

$$B^{-1} = (\sigma_n^2 I + K_{nm}K_{mm}^{-1}K_{mn})^{-1} = \frac{I}{\sigma_n^2} - \frac{K_{nm}A^{-1}K_{mn}}{\sigma^4},$$

which allows for computing $y^T B^{-1} y$ in $O(nm)$.

Similarly, we can compute the gradient in $O(nm^2)$. In order to do so, we need to rewrite every trace $\mathrm{tr}(M_{nm}M_{mm}M_{mn})$, where $M_{kl} \in \mathbb{R}^{k\times l}$, in the form $\mathrm{tr}(M_{mm}M_{mn}M_{nm})$, which is computed in $O(nm^2)$, and use the derived formulas for B^{-1}.

Now let's try to compute the second order derivatives.

$$\frac{\partial^2 F_V}{\partial\theta_j\partial\theta_i} = \frac{\partial}{\partial\theta_j}\left(\frac{\partial F_V}{\partial\theta_i}\right) = \frac{1}{2}\frac{\partial}{\partial\theta_j}\left(-\mathrm{tr}\left(B^{-1}\frac{\partial B}{\partial\theta_i}\right) + y^T B^{-1}\frac{\partial B}{\partial\theta_i}B^{-1}y - \frac{1}{\sigma_n^2}\mathrm{tr}\left(\frac{\partial K_{nn}}{\partial\theta_i} - \frac{\partial B}{\partial\theta_i}\right)\right) =$$

$$= \frac{1}{2}\left(\mathrm{tr}\left(B^{-1}\frac{\partial B}{\partial\theta_j}B^{-1}\frac{\partial B}{\partial\theta_i} - B^{-1}\frac{\partial^2 B}{\partial\theta_j\partial\theta_i}\right) + y^T\left(B^{-1}\frac{\partial^2 B}{\partial\theta_j\partial\theta_i}B^{-1} - 2B^{-1}\frac{\partial B}{\partial\theta_j}B^{-1}\frac{\partial B}{\partial\theta_i}B^{-1}\right)y - \right.$$
$$\left. -\frac{1}{\sigma_n^2}\mathrm{tr}\left(\frac{\partial^2 K_{nn}}{\partial\theta_j\partial\theta_i} - \frac{\partial^2 B}{\partial\theta_j\partial\theta_i}\right)\right),$$

where

$$\frac{\partial^2 B}{\partial\theta_j\partial\theta_i} = \frac{\partial}{\partial\theta_j}\left(\frac{\partial K_{nm}}{\partial\theta_i}K_{mm}^{-1}K_{mn} - K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_i}K_{mm}^{-1}K_{mn} + K_{nm}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial\theta_i}\right) =$$

$$= \frac{\partial^2 K_{nm}}{\partial\theta_j\partial\theta_i}K_{mm}^{-1}K_{mn} + K_{nm}K_{mm}^{-1}\frac{\partial^2 K_{mn}}{\partial\theta_j\partial\theta_i} - \frac{\partial K_{nm}}{\partial\theta_i}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_j}K_{mm}^{-1}K_{mn} -$$
$$-K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_j}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial\theta_i} + \frac{\partial K_{nm}}{\partial\theta_j}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial\theta_i} + \frac{\partial K_{nm}}{\partial\theta_i}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial\theta_j}$$
$$-\frac{\partial K_{nm}}{\partial\theta_j}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_i}K_{mm}^{-1}K_{mn} + K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_j}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_i}K_{mm}^{-1}K_{mn}$$
$$-K_{nm}K_{mm}^{-1}\frac{\partial^2 K_{mm}}{\partial\theta_j\partial\theta_i}K_{mm}^{-1}K_{mn} + K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_i}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_j}K_{mm}^{-1}K_{mn} -$$
$$-K_{nm}K_{mm}^{-1}\frac{\partial K_{mm}}{\partial\theta_i}K_{mm}^{-1}\frac{\partial K_{mn}}{\partial\theta_j}.$$

2.1.2 Stochastic variational inference

The method discussed here was proposed in [2]. The method doesn't provide a way to choose the positions of inducing points. It provides a way to find the predictive distribution and optimize hyper-parameters for large datasets.

For using stochastic variational inference, we have to provide a lower bound for the marginal likelihood, that factorizes over the training examples. To obtain such an ELBO (evidence lower bound) two ancillary lower bounds are found.

By applying the Jensen inequality we obtain

$$\log p(y|u) = \log \left(\int p(y|f)p(f|u)du \right) \geq \int \log(p(y|f))p(f|u)du = L_1.$$

As $p(y|f)$ factorizes over examples we obtain

$$\exp(L_1) = \prod_{i=1}^{n} \mathcal{N}(y_i|\mu_i, \sigma_n^2) \exp\left(-\frac{1}{2\sigma_n^2} \tilde{K}_{ii} \right).$$

Note that

$$\log p(y|u) - L_1 = \mathrm{KL}\left(p(f|u) \;||\; p(f|u, y) \right).$$

Using the lower bound L_1 we obtain a lower bound for the marginal likelihood

$$\log p(y) = \log \left(\int p(y|u)p(u)du \right) \geq \log \left(\int \exp(L_1)p(u)du \right) = L_2.$$

With some algebraic manipulations we obtain the following expression for L_2

$$L_2 = \log \mathcal{N}(y|0, K_{nm}K_{mm}^{-1}K_{mn} + \sigma_n^2 I) - \frac{1}{2\sigma_n^2}\mathrm{tr}(\tilde{K}).$$

This is exactly the expression for the lower bound, used in the method, described in the section 2.1.1 for the optimal approximating distribution $q(u) = \mathcal{N}(u|\hat{u}, \Lambda^{-1})$, where

$$\Lambda = \frac{1}{\sigma_n^2} K_{mm}^{-1} K_{mn} K_{nm} K_{mm}^{-1} + K_{mm}^{-1},$$

$$\hat{u} = \frac{1}{\sigma_n^2} \Lambda^{-1} K_{mm}^{-1} K_{mn} y.$$

In the method, described in section 2.1.1, this lower bound is being maximized over the kernel hyper-parameters and the optimal distribution $q(u)$ is used for making predictions at unseen points x as follows

$$\mathbb{E}f(x) = K_{xm}K_{mm}^{-1}\hat{u},$$

$$\mathrm{cov}(f(x), f(x')) = k(x, x') - K_{xm}K_{mm}^{-1}K_{mx'} + K_{xm}K_{mm}^{-1}\Lambda^{-1}K_{mm}^{-1}K_{mx'}.$$

Unfortunately, evaluating Λ takes $O(nm^2)$ operations and thus this method cannot be applied to big datasets. To overcome this limitation, we will use stochastic optimization to find the approximate optimal distribution $q(u)$ and to optimize for hyper-parameters.

Let the variational distribution q be normal with mean μ and covariance matrix Σ. The final ELBO is derived as follows

$$\log p(y) \geq \int \left(L_1 + \log p(u) - \log q(u)\right) q(u) du = L_3.$$

This lower bound factorizes over the examples

$$L_3 = \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma_n^2) - \frac{1}{2\sigma_n^2} \tilde{K}_{ii} - \frac{1}{2} \mathrm{tr}(\frac{1}{\sigma_n^2} \Sigma K_{mm}^{-1} k_i k_i^T K_{mm}^{-1}) \right) - \mathrm{KL}\left(q(u) \mid\mid p(u)\right) =$$

$$= \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \mu, \sigma_n^2) - \frac{1}{2\sigma_n^2} \tilde{K}_{ii} - \frac{1}{2} \mathrm{tr}(\Sigma \Lambda_i) \right) -$$
$$- \frac{1}{2} \left(\log \frac{|K_{mm}|}{|\Sigma|} - m + \mathrm{tr}(K_{mm}^{-1} \Sigma) + \mu^T K_{mm}^{-1} \mu \right),$$

where $\Lambda_i = \frac{1}{\sigma_n^2} K_{mm}^{-1} k_i k_i^T K_{mm}^{-1}$, and k_i is the i-th column of the matrix K_{mn}.

In stochastic variational inference natural gradients are used to maximize the ELBO. The canonical parameters for the normal distribution $q(u)$ are

$$\eta_1 = \Sigma^{-1} \mu, \quad \eta_2 = -\frac{1}{2} \Sigma^{-1}.$$

The expectation parameters are

$$\beta_1 = \mu, \quad \beta_2 = \mu \mu^T + \Sigma.$$

In the exponential family the natural gradients are equal to the gradients with respect to expectation parameters. To find these gradients we first reparametrise the ELBO

$$L_3(\beta_1, \beta_2) = \sum_{i=1}^{n} \left(\log \mathcal{N}(y_i | k_i^T K_{mm}^{-1} \beta_1, \sigma_n^2) - \frac{1}{2\sigma_n^2} \tilde{K}_{ii} - \frac{1}{2} \mathrm{tr}((\beta_2 - \beta_1 \beta_1^T) \Lambda_i) \right) -$$

$$- \frac{1}{2} \left(\log |K_{mm}| - \log |\beta_2 - \beta_1 \beta_1^T| - m + \mathrm{tr}(K_{mm}^{-1} (\beta_2 - \beta_1 \beta_1^T)) + \beta_1^T K_{mm}^{-1} \beta_1 \right).$$

Differentiating with respect to expectation parameters we obtain

$$\frac{\partial L_3}{\partial \beta_1} = -\frac{1}{\sigma_n^2} \sum_{i=1}^{n} \left(K_{mm}^{-1} k_i y_i \right) + \Sigma^{-1} \mu, \tag{8}$$

$$\frac{\partial L_3}{\partial \beta_2} = \frac{1}{2} \left(-\sum_{i=1}^{n} (\Lambda_i) + \Sigma^{-1} - K_{mm}^{-1} \right). \tag{9}$$

The natural gradient descent updates of these parameters are

$$\eta_{1(t+1)} = \Sigma^{-1}_{(t+1)}\mu_{(t+1)} = \Sigma^{-1}_{(t)}\mu_{(t)} + \ell\left(\frac{1}{\sigma_n^2}K^{-1}_{mm}K_{mn}y - \Sigma^{-1}_{(t)}\mu_{(t)}\right),$$

$$\eta_{2(t+1)} = -\frac{1}{2}\Sigma^{-1}_{(t+1)} = -\frac{1}{2}\Sigma^{-1}_{(t)} + \ell\left(-\frac{1}{2}\Lambda + \frac{1}{2}\Sigma^{-1}_{(t+1)}\right),$$

where ℓ is the step length. It's easy to see, that if $\ell = 1$ the method converges to the optimal distribution $q(u)$ in one iteration. Unfortunately, we can not directly compute the updates described above, because the computational complexity of computing the matrix Λ is $O(nm^2)$. We will use approximations to the natural gradients, obtained by considering the data points individually or in batches. The formulas for these approximations can be obtained from equalities 8, 9.

Finally, we need to find the derivatives of the ELBO with respect to kernel hyper-parameters θ apart from σ_n^2

$$\frac{\partial L_3}{\partial \theta} = \sum_{i=1}^{n}\left[\frac{1}{\sigma_n^2}(y_i - k_i^T K^{-1}_{mm}\mu)\left(\frac{\partial k_i^T}{\partial \theta}K^{-1}_{mm} - k_i^T K^{-1}_{mm}\frac{\partial K_{mm}}{\partial \theta}K^{-1}_{mm}\right)\mu + \right.$$

$$+\frac{1}{2\sigma_n^2}\left(-\frac{\partial K_{nn}}{\partial \theta} + \frac{\partial K_{nm}}{\partial \theta}K^{-1}_{mm}K_{mn} + K_{nm}K^{-1}_{mm}\frac{\partial K_{mm}}{\partial \theta}K^{-1}_{mm}K_{mn} + K_{nm}K^{-1}_{mm}\frac{\partial K_{mn}}{\partial \theta}\right)_{ii} +$$

$$\left. +\frac{1}{\sigma_n^2}\mathrm{tr}\left(\Sigma\left(K^{-1}_{mm}\frac{\partial K_{mm}}{\partial \theta}K^{-1}_{mm}k_i k_i^T K^{-1}_{mm} - K^{-1}_{mm}\frac{\partial k_i}{\partial \theta}k_i^T K^{-1}_{mm}\right)\right)\right] -$$

$$-\frac{1}{2}\mathrm{tr}\left(K^{-1}_{mm}\frac{\partial K_{mm}}{\partial \theta}\right) + \frac{1}{2}\mathrm{tr}\left(\Sigma K^{-1}_{mm}\frac{\partial K_{mm}}{\partial \theta}K^{-1}_{mm}\right) + \frac{1}{2}\mu^T K^{-1}_{mm}\frac{\partial K_{mm}}{\partial \theta}K^{-1}_{mm}\mu,$$

and for σ_n we have the same formula plus the following correction

$$\sum_{i=1}^{n}\left(-\frac{1}{\sigma_n} + \frac{1}{\sigma_n^3}(k_i^T K^{-1}_{mm}\mu - y_i)^2 + \frac{1}{\sigma_n^3}\tilde{K}_{ii} + \frac{\mathrm{tr}(\Sigma\Lambda_i)}{\sigma_n}\right).$$

Now, we can optimize the kernel hyper-parameters and the noise variance alongside the variational parameters.

We can also maximize the L_3 with procedures, other than stochastic gradient descent. However, in most of the effective optimization methods we can't use natural gradients, because they are not necesserily a descending direction. Thus, we have to use the usual gradients. However, there is a problem with this approach as well. The steps in the direction of the antigradient does not guarantee that the updated covariance Σ is positive definite.

To solve this problems, we use Choletsky decomposition L_Σ of Σ and optimize L_3 with respect to it.

$$L_3(L_\Sigma, \mu) = \sum_{i=1}^{n}\left(\log \mathcal{N}(y_i | k_i^T K^{-1}_{mm}\mu, \sigma_n^2) - \frac{1}{2\sigma_n^2}\tilde{K}_{ii} - \frac{1}{2}\mathrm{tr}(L_\Sigma L_\Sigma^T \Lambda_i)\right) -$$

$$-\frac{1}{2}\left(\log\frac{|K_{mm}|}{|L_\Sigma L_\Sigma^T|} - m + \mathrm{tr}(K_{mm}^{-1}L_\Sigma L_\Sigma^T) + \mu^T K_{mm}^{-1}\mu\right) =$$

$$= \sum_{i=1}^{n}\left(\log\mathcal{N}(y_i|k_i^T K_{mm}^{-1}\mu, \sigma_n^2) - \frac{1}{2\sigma_n^2}\tilde{K}_{ii} - \frac{1}{2}\mathrm{tr}(L_\Sigma^T \Lambda_i L_\Sigma)\right) -$$

$$-\frac{1}{2}\left(\log|K_{mm}| - 2\sum_{j=1}^{m}\log(L_\Sigma)_{jj} - m + \mathrm{tr}(L_\Sigma^T K_{mm}^{-1} L_\Sigma) + \mu^T K_{mm}^{-1}\mu\right)$$

The gradients with respect to μ and L_σ are given by

$$\frac{\partial L_3}{\partial \mu} = \sum_{i=1}^{n}\left(\Lambda_i\mu - \frac{y_i}{\sigma_n^2}K_{mm}^{-1}k_i\right) + K_{mm}^{-1}\mu,$$

$$\frac{\partial L_3}{\partial L_\Sigma} = -\sum_{i=1}^{n}\Lambda_i L_\Sigma + \begin{pmatrix} \frac{1}{(L_\Sigma)_{11}} & 0 & \dots & 0 \\ 0 & \frac{1}{(L_\Sigma)_{22}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{(L_\Sigma)_{mm}} \end{pmatrix} - K_{mm}^{-1}L_\Sigma.$$

3 Experiments

In this section the results of the numerical experiments are provided. All of the provided plots has a title, that tells the number of training points n, the number of features d and the number of inducing points m. The title also tells the name of the dataset.

The methods were compared on variaus datasets. Some of them are generated from a gaussian process and others are real. The R^2-score on a test set was used as a quality metric.

The squared exponential kernel was used in all the experiments.

3.1 Variations of the stochastic variational inference method

In this section we compare several variations of the stochastic variational inference method.

The first variation is denoted by `svi-natural`. It is the method described in [2]. It uses stochastic gradient descent with natural gradients for minimizing the ELBO with respect to the variational parameters, and usual gradients with respect to kernel hyperparameters.

The methods `svi-L-BFGS-B` and `svi-FG` use the full (non-stochastic) ELBO from the same article [2] and minimize it with L-BFGS-B and gradient descent respectively. These methods use Cholesky factorization (see 2.1.2) for the variational parameters.

Finally, the `svi-SAG` method to minimize the ELBO. This method also uses Cholesky factorization.

We will compare the methods on datasets, generated from some gaussian process and on real data.

The results on small and medium datasets are shown in fig. 2 and fig. **??**.

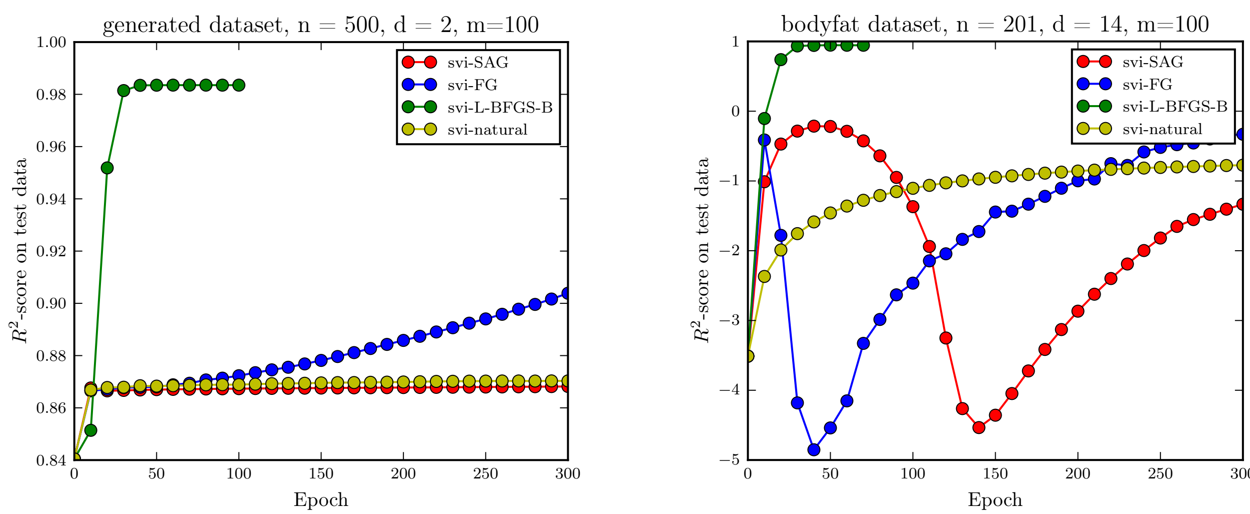

Рис. 2: Svi methods' performance on small datasets

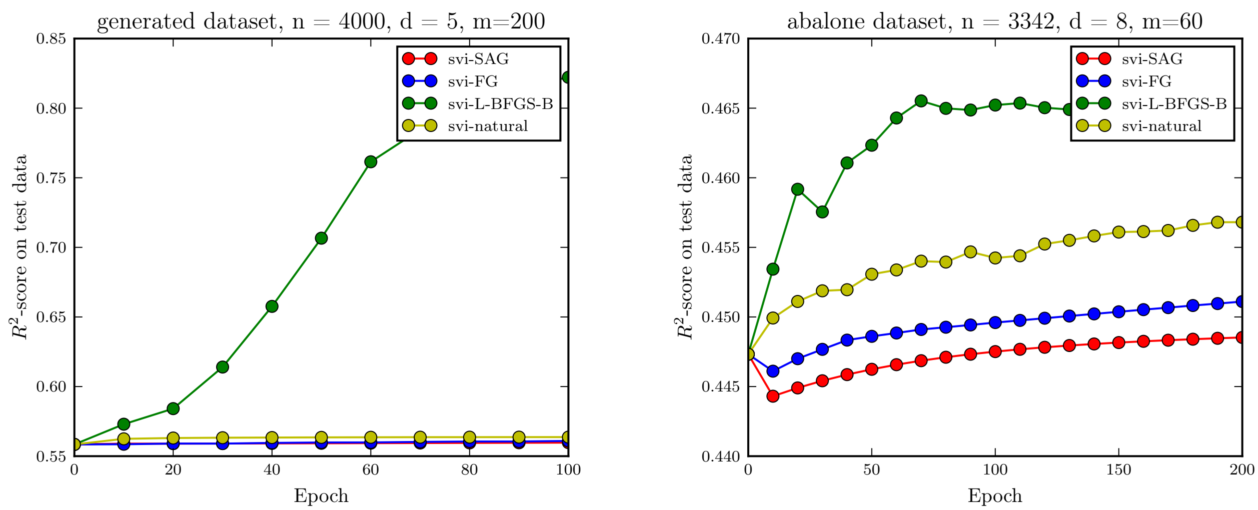

Рис. 3: Svi methods' performance on medium datasets

3.2 Comparison of stochastic and non-stochastic variational inference methods

In this section we compare the `vi-means` method with `svi-L-BFGS-B`. The `vi-means` method is a variation of the method, described in section 2.1.1. It does not optimize for the inducing point positions and does uses `L-BFGS-B` to maximize the ELBO.

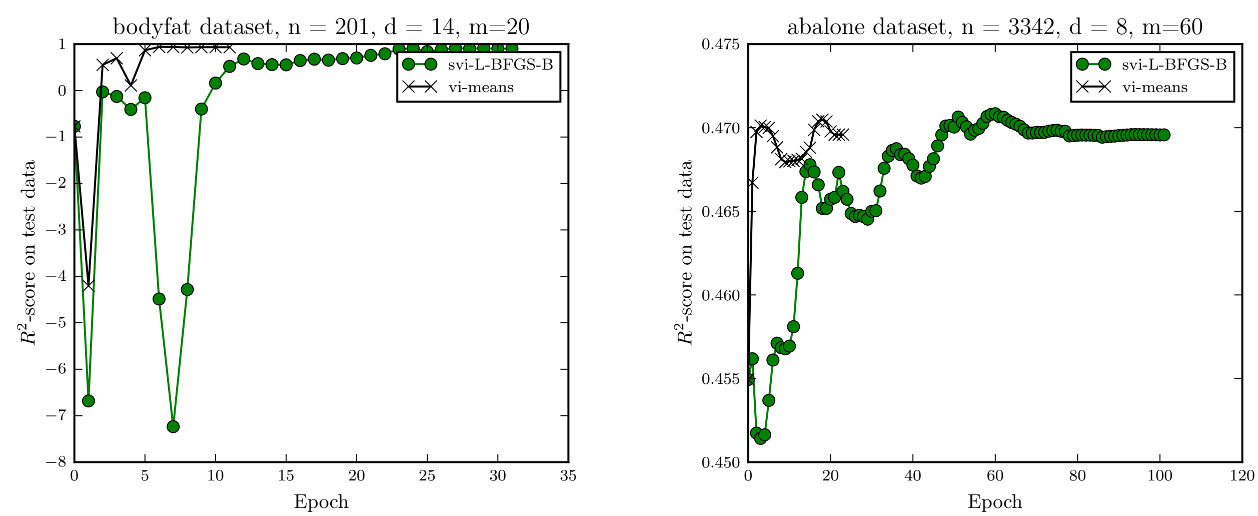

Рис. 4: Method's performance on small and medium datasets

We can see, that `vi-means` beats it's oponent in all the experiments. One could expect these results, because `vi-means` optimizes the exact same functional as it's oponent, but

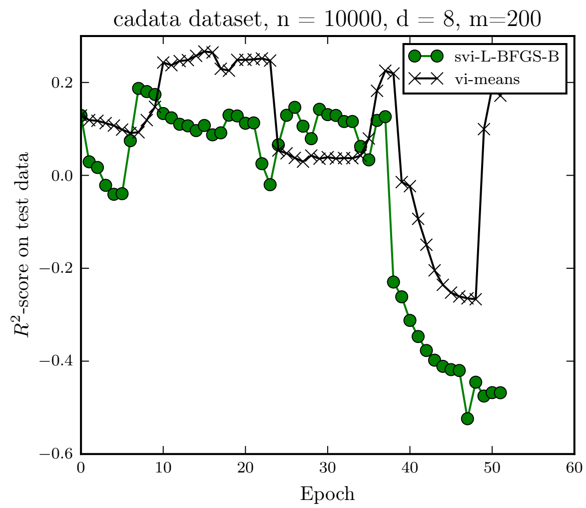

Рис. 5: Method's performance on a big dataset

it uses exact optimal values for some of the parameters. Thus, on moderate problems the `vi-means` method beats all the discussed `svi` variations.

Finally, we will compare `vi-means` with stochastic `svi-natural` an `svi-SAG` on a big dataset. The results can be found in fig. 6.

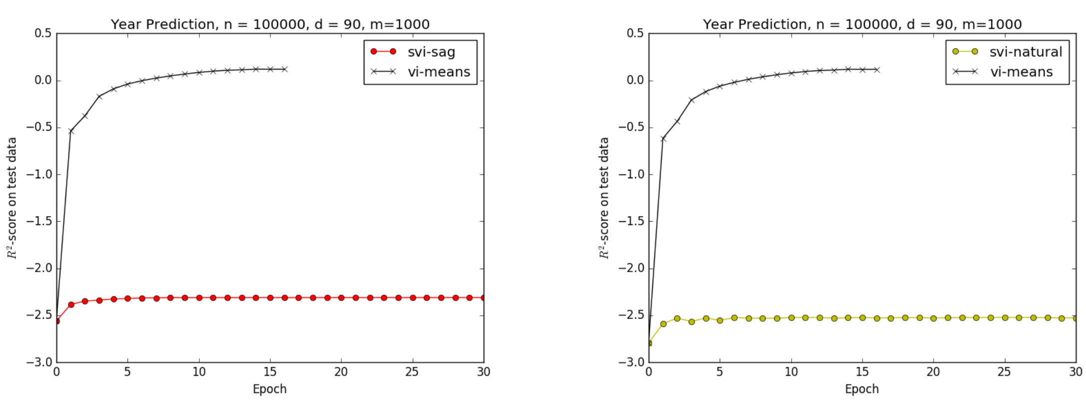

Рис. 6: vi and svi methods comparison on a big dataset

3.3 Variations of variational inference method

In this section we compare several variations of the stochastic variational inference method. The method itself is described in section 2.1.1. We compare two different optimization methods for minimizint the Titsias's ELBO.

The first variation is denoted by `means-PN`. It uses Projected-Newton method for minimizing the ELBO. The second variation is denoted by `means-L-BFGS-B` and uses L-BFGS-B optimization method.

The `means-PN` uses finite-difference approximation of the hessian. It also makes hessian-correction in order to make it simmetric positive-definite.

We compare the methods on several different datasets. The results on a small and medium datasets can be found in fig. **??**. The results on a biger dataset can be found in fig. **??**

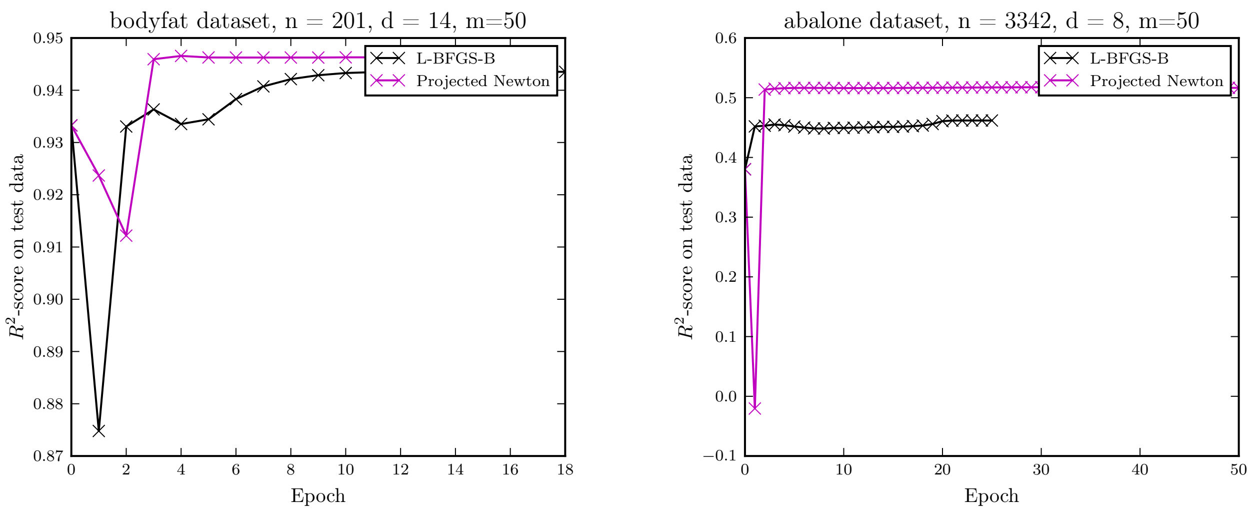

Рис. 7: Method's performance on small and medium datasets

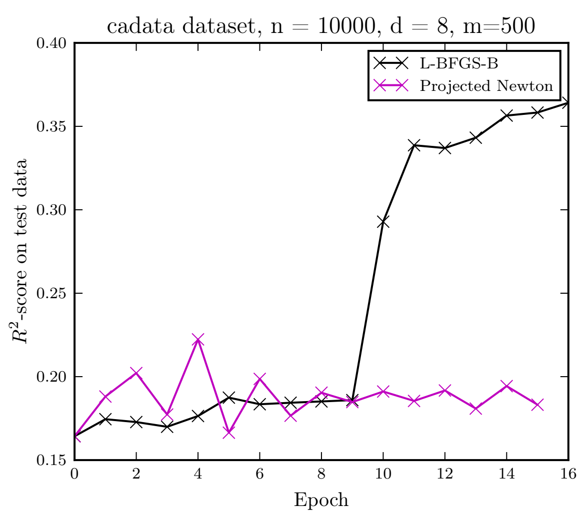

Рис. 8: Method's performance on a bigger dataset

Literature

[1] Titsias M. K. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In: *International Conference on Artificial Intelligence and Statistics*, pp. 567–574.

[2] Hensman J., Fusi N., Lawrence D. (2013). Gaussian Processes for Big Data. In: *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*.