Requirements for the projects

Successful project includes the following components:

- Project report at least 10 pages long that includes relevant links to the state of the art, review of the state of the art, problem statement, and obtained results. Report should be provided as pdf document.
- Presentation given at the last week of the course (precise time of talks will be identified later), presentation report length is 15 minutes (10 minutes talk + 5 minutes questions), the presentation should be 7-15 slides long.
- Source code available for TAs.

The main assessment criteria and maximum points:
1) The general literacy and style of the report — 10%;
2) Analytical/scientific methods and approaches — 20%;
3) Depth of the subject understanding— 45%;
4) The presentation and answers to questions — 25%.

We encourage you to format the report as a scientific article, so you can later submit obtained results to a journal dedicated to publication of student-level articles (see e.g. http://jmlda.org/journal).

We don't limit number of students involved in one project, but we expect for each project to provide in a report a description of personal contributions of each student (so, we recommend to form teams of size 2-3 people).

Assessment criteria for the final project are similar to that of other Skoltech courses.

The list of the problems is specified below. You can choice another problem, but this problem should be connected with Bayesian machine learning, so an approvement by TAs or the instructor is required.

Problem 1: HMM

Annotation: HMM is a powerful model to infer some latent state (often a class) by analyzing time-series measurements.

Task: Your task would be to:

- implement Hidden Markov Model algorithm and learning primitives
- download and preprocess data from Actitracker Dataset (http://www.cis.fordham.edu/wisdm/dataset.php)
- train your model
- perform inference and analyze performance of your model

This dataset contains data collected through controlled, laboratory conditions. If you are interested in "real world" data, please consider usage of an alternative real data set.

You will need to create an animated visualisation of HMM inference (latent variables, observable variables and their connections). You need to show how information “flows through model at inference time”.

Problem 2: Cokriging

Annotation: Engineering problems often involve variable fidelity: in such a case variable fidelity data come from sources of different fidelity. For example, we obtain high fidelity data from experiments in wind tunnel and low fidelity data from CFD numerical experiments.

Common way for construction of regression models using variable fidelity data is application of so-called cokriging approach, see section 3 of the article https://www.researchgate.net/publication/315824185_Large_scale_variable_fidelity_surrogate_modeling for more details of model and its’ usage.

Task: The goal of the project is implementation of cokriging on the base of GPy package https://github.com/SheffieldML/GPy. For testing of the approach one can use datasets presented at https://www.gitlab.com/JohnDoe1989/VariableFidelityData (both data and some scripts in Matlab are available, see more details in the article https://www.researchgate.net/publication/309321717_Minimax_Error_of_Interpolation_and_Optimal_Design_of_Experiments_for_Variable_Fidelity_Data)

Problem 3: Bayesian optimization

Annotation: The goal of Bayesian optimization is to find optimal value of the function by construction of regression model on the base of given data, and then optimization of this regression model instead of a heavy blackbox function. Bayesian optimization is widely used for example to estimate hyperparameters of machine learning algorithm e.g. for selection of gamma parameter for SVM. See more on Bayesian optimization in the article https://arxiv.org/pdf/1206.2944.pdf and reference therein.

Popular approach for construction of nonlinear regression models in Bayesian optimization is Gaussian process regression, as this approach provides uncertainty estimates of the model.

Task: You should apply Bayesian optimization for one of two problems: finding parameters of Universe that provide the most reliable estimate of red shifts of supernovas, and finding optimal parameters of SVM using cross-validation. See
https://gitlab.com/JohnDoe1989/VariableFidelityData/blob/master/data/supernova/generate_supernova_data.ipynb
and
https://gitlab.com/JohnDoe1989/VariableFidelityData/blob/master/data/svm/generate_svm_data.ipynb
for two examples of the blackboxes for the problems introduced above.
You can read about the problem in more details at
https://www.researchgate.net/publication/309321717_Minimax_Error_of_Interpolation_and_Optimal_Design_of_Experiments_for_Variable_Fidelity_Data
and https://arxiv.org/abs/1603.06288.

We expect that you will use some open libraries like https://github.com/SheffieldML/GPy that implement Gaussian process regression. Criterions to start with are Expected Improvement and Maximum Variance as well as current minimum of the regression model and their combinations.

Problem 4: Selection of prior for Bayesian neural network

Annotation: One way for construction of Neural Networks with good generalization ability is to use prior distributions for parameters of Neural Network. Introduction of Bayesian approach provides regularization for parameters of Neural networks thus avoiding redundantly complex models and big values of parameters.

Task: An example of usage of Bayesian approach for deep learning is available at http://twiecki.github.io/blog/2016/06/01/bayesian-deep-learning/
In this project you should try different families of probabilistic distributions as prior distributions and make a conclusion on what prior we should use for this particular problem. You should try at least three different priors e.g. Gaussian, Laplace and Cauchy. You should provide strong experimental evidence for your conclusions. To generate various data sets use functions `make_moons, make_circles` from scikit-learn library.

Problem 5: Inverting complex functions with MCMC
(Student: Dima Mironov)
Annotation: To perform analysis of complex images or videos a lot of times simple labels aren't enough. To "understand" scene we need to explain it, using some more flexible language. Objects in images are not random, but can be interpreted as probabilities on some manifold which were transformed by lighting and projection. Solving complex inverse graphics problem can be done with probabilistic methods. (https://mrkulk.github.io/www_cvpr15/1999.pdf , https://github.com/mrkulk/MIT-Picture)

Task: Extend "Picture - probabilistic scene description language" to perform motion capture from video, by using "Pose Estimation demo" and motion capture data for a prior on movement dynamics. Evaluate algorithm on HumanEva dataset (http://humaneva.is.tue.mpg.de/), or its sub-sample.

Problem 6: Compare PyStan and PyMC3 for some complex problem

Annotation: There are a number of modern libraries dedicated to probabilistic programming, in particular for Bayesian inference. Among them the most established are PyStan, PyMC3.

Task: In this project you should compare how two libraries solve the same complex problem related to Bayesian inference. You should compare quality of obtained model and time required for construct and evaluation of the model.

We propose two “complex” problems. The first problem is Bayesian Gaussian Mixture Model described at seminar, as a dataset you can use Iris dataset or any other dataset dedicated to clustering on your choice. The second problem is Gaussian process classification, see a chapter http://www.gaussianprocess.org/gpml/chapters/RW3.pdf for an overview. In this case you can also use for example Iris dataset.

Problem 7: Comparing model selection criteria

Annotation: One of the advantages of Bayesian approach is an ability to select models on the base of obtained Bayesian estimates. We propose to apply different model selection criteria for selection of number of components in Bayesian Gaussian Mixture model described in C.Bishop book, p. 483-485.

Task: We propose to apply different model selection criteria for selection of number of components in Bayesian GMM. You should compare different model selection criteria (including correctly penalized variational lower bound available in Bishop and another two criteria e.g. AIC, http://www4.ncsu.edu/~shu3/Presentation/AIC.pdf (presentation) BIC, https://projecteuclid.org/euclid.aos/1176344136, DIC, http://onlinelibrary.wiley.com/doi/10.1111/rssb.12062/pdf, or WIAC, http://www.jmlr.org/papers/volume14/watanabe13a/watanabe13a.pdf).

Select a dataset suitable for clustering among presented at UCI repository https://archive.ics.uci.edu/ml/datasets.html?format=&task=clu&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=table
For implementation we encourage you to use PyMC3 or an update of your code for homework 2.

Problem 8: Adding Nonparametric models to PyMC3

Annotation: Nonparametric methods can be used to automatically infer models of adequate size and complexity without thinking about Bayesian model comparison. You will be implementing these models into PyMC3 to extend the functionality and make it more useful for Bayesian Statistics and Inference. The nonparametric Bayesian methods are powerful since they can be derived by starting with a finite parameter and taking the limit of parameters up to infinity. Nonparametric Bayesian Method are described in brief in the class in MIT by Lorenzo Rosasco (http://www.mit.edu/~9.520/spring11/slides/class18_dirichelet.pdf).

Task: Implement nonparametric Bayesian Methods for PyMC3 library. Code should be of the industrial quality (efficient, readable, documented, tested, examples in jupyter notebooks).
You need to implement “Dirichlet Process” probability distribution, and at least two of the following:

1. Dirichlet process mixtures
2. Chinese Restaurant Process
3. Hierarchical Dirichlet Processes
4. Indian Buffet Processes
5. Dirichlet Diffusion Trees

6. Infinite Hidden Markov Models

You will be assisted with code quality control assessment.