Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Software Engineering in
Energy Industry Department

Bayesian Statistics with Python

Syllabus

Catalog Description

Higher education level	*Second (Graduate)*
Knowledge field	*Information Technologies*
Profession	*121 Software Engineering*
Curriculum	*Software Engineering of Intelligent Cyber-Physical Systems in Energy Industry*
Course status	*Elective*
Form of training	*Full-time*
Grade, term	*First grade, spring semester*
Credits (hours)	*4 credits / 120 hours (full time: 36 hours of lectures, 28 hours of practice, 66 hr of individual assignments)*
Term control	*Exam, modular test*
Schedule	*http://schedule.kpi.ua/*
Teaching language	*Ukrainian/English*
Instructors	Lecturer: *DSc. (Econ), professor Andrii Sihaiov* Seminars: Laboratory work: *Andrii Sihaiov*
URL	

Course Program

1 Course description, aim, subject, and expected outcomes

Why future specialist should study this course?

Bayesian methods of inference are deeply natural and extremely powerful. However, most discussions of Bayesian inference rely on intensely complex mathematical analyses and artificial examples, making it inaccessible to anyone without a strong mathematical background. This course, though, introduces Bayesian inference from a computational perspective, bridging theory to practice–freeing you to get results using computing power.

Our course illuminates Bayesian inference through probabilistic programming with the powerful PyMC language and the closely related Python tools NumPy, SciPy, and Matplotlib. Using this approach, you can reach effective solutions in small increments, without extensive mathematical intervention.

We begin by introducing the concepts underlying Bayesian inference, comparing it with other techniques and guiding you through building and training your first Bayesian model. Next, we introduce PyMC through a series of detailed examples and intuitive explanations that have been refined after extensive user feedback. You'll learn how to use the Markov Chain Monte Carlo algorithm, choose appropriate sample sizes and priors, work with loss functions, and apply Bayesian inference in domains ranging from finance to marketing. Once you've mastered these techniques, you'll constantly turn to this guide for the working PyMC code you need to jumpstart future projects.

Course Aim. *To familiarize students with probabilistic programming and bayesian inference.*

Course Subject. *Coverage includes:*

- *Learning the Bayesian "state of mind" and its practical implications*

- *Understanding how computers perform Bayesian inference*
- *Using the PyMC Python library to program Bayesian analyses*
- *Building and debugging models with PyMC*
- *Testing your model's "goodness of fit"*
- *Opening the "black box" of the Markov Chain Monte Carlo algorithm to see how and why it works*
- *Leveraging the power of the "Law of Large Numbers"*
- *Mastering key concepts, such as clustering, convergence, autocorrelation, and thinning*
- *Using loss functions to measure an estimate's weaknesses based on your goals and desired outcomes*
- *Selecting appropriate priors and understanding how their influence changes with dataset size*
- *Overcoming the "exploration versus exploitation" dilemma: deciding when "pretty good" is good enough*
- *Using Bayesian inference to improve A/B testing*
- *Solving data science problems when only small amounts of data are available*

Expected Outcomes.

Professional Competencies.

PC 11. Ability to design and develop software systems using methods of intelligent data analysis.

Program's Learning Outcomes.

PLO 17. Collect, analyze, evaluate information necessary for solving scientific and applied problems, using scientific and technical literature, databases and other sources.

PLO 19. Be able to design and develop software systems using methods of intelligent data analysis.

2 Course prerequisites (Where the course fits into our curriculum)

The course is taken in spring semester of final year. Basic knowledge of probabilty theory and Python are prerequisites. There is no required course that has this course as a prerequisite.

3 Course contents

1. The Philosophy of Bayesian Inference.

2. A Little More on PyMC.

3. Opening the Black Box of MCMC.

4. The Greatest Theorem Never Told.

5. Would You Rather Lose an Arm or a Leg?

6. Getting Our Priorities Straight.

7. Bayesian A/B Testing.

4 Course textbooks and materials

Required reading:

Davidson-Pilon, C. Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference: Upper Saddle River, NJ: Addison-Wesley Professional, 2015. 256 c. URL: http://libgen.rs/book/index.php?md5=51d73c88197e1de5d8128f839a61d0e2

Optional reading:

1. *Kurt, W. Bayesian Statistics the Fun Way: Understanding Statistics and Probability with Star Wars, LEGO, and Rubber Ducks: San Francisco, CA: No Starch Press, 2019. 256 c. URL: http://libgen.rs/book/index.php?md5=BFE1699AAC3B103EDF0FF53169BAA645*
2. *Martin, O. Bayesian Analysis with Python: Introduction to statistical modeling and probabilistic programming using PyMC3 and ArviZ: Birmingham, UK: Packt Publishing, 2018. 356 c. URL: http://libgen.rs/book/index.php?md5=9608A3B42E5D4C1875DCA1C60506B1D2*

Educational Content

5 Pedagogical advice

1. *The Philosophy of Bayesian Inference*
   - 1.1. *Introduction*
     - 1.1.1. *The Bayesian State of Mind*
     - 1.1.2. *Bayesian Inference in Practice*
     - 1.1.3. *Are Frequentist Methods Incorrect?*
     - 1.1.4. *A Note on “Big Data”*
   - 1.2. *Our Bayesian Framework*
     - 1.2.1. *Example: Mandatory Coin-Flip*
     - 1.2.2. *Example: Librarian or Farmer?*
   - 1.3. *Probability Distributions*
     - 1.3.1. *Discrete Case*
     - 1.3.2. *Continuous Case*
     - 1.3.3. *But What Is λ?*
   - 1.4. *Using Computers to Perform Bayesian Inference for Us*
     - 1.4.1. *Example: Inferring Behavior from Text-Message Data*
     - 1.4.2. *Introducing Our First Hammer: PyMC*
     - 1.4.3. *Interpretation*
     - 1.4.4. *What Good Are Samples from the Posterior, Anyway?*
   - 1.5. *Conclusion*
   - 1.6. *Appendix*
     - 1.6.1. *Determining Statistically if the Two s Are Indeed Different?*
     - 1.6.2. *Extending to Two Switchpoints*
   - 1.7. *Exercises*
     - 1.7.1. *Answers*
   - 1.8. *References*
2. *A Little More on PyMC*

6 Individual Assignments

Students expected to spend 3-4 hours a week outside of class on course material.

Course Rules and Assessment Policy

7 Course study rules

Students receive points relative to the correct and timely completion of coursework. The total grade consists of: 1) laboratories (programming assignments) 60%, 2) final exam 40%.
Presently there are three programming assignments, each worth up to 20% of the total grade. Student have to submit correctly fulfilled assignment during fortnight period from the date of give out to obtain full score for it, otherwise penalty points are applied but not more than 40% of the total score for laboratory work.

8 Assessment policy

How students are assessed: modular test, programming assignments
Calendar control: conducted twice a term to monitor the current state of compliance with the requirements of the syllabus.
Term assessment: final exam
Admission condition of term assessment: all programming assignment submission, start score not less than 40 points.
Exam scores map to the course grade according to the table:

Score	*Grade*
100-95	Excellent
94-85	Very Good
84-75	Good
74-65	Satisfactory
64-60	Sufficient
Less than 60	Unsatisfactory
Conditions for exam admission not met	Not allowed

9 Additional topics

- *Exam questions (see appendix).*

Syllabus:

Developed by Software Engineering in Energy Industry Department Professor, Sc. D. Andrii Sihaiov

Approved by Software Engineering in Energy Industry Department (minutes #28 on May 15, 2023)

Endorsed by Methodical Commission of Heat Power Faculty (minutes #9 on May 26, 2023)