610 course webpage Documentation

Release 2020

Varun Chandola

Nov 17, 2020

Contents:

CHAPTER 1

Syllabus

Computer Science and Engineering, University at Buffalo

Fall Semester 2020

1.1 Instructors

- Varun Chandola (lead instructor; chandola[at]buffalo.edu)

Note: Students are strongly encouraged to use the Piazza's private messaging option to contact the intructors to ensure that the messages are dealt with promptly.

1.2 Meeting times and locations

	Day	Time	Location
Lectures (in-person)	Tuesdays and Thursdays	11.10 AM - 12.25 PM	Talbert 107
Lectures (online/live)	Tuesdays and Thursdays	11.10 AM - 12.25 PM	Online (UB Panopto/zoom-lectures)
Lectures (recorded)			Online (UB Panopto)
Office Hours	Wednesdays	12.00 PM - 2.00 PM	Online (zoom-oh)

Attention: This class will taught in a **HyFlex Mode**. The class content covered during the in-person lectures will be available online through UB Panopto, both live and recorded. A separate session on zoom will be available during the lectures for asking questions. All office hours will be conducted remotely on zoom.

1.3 Prerequisites

CSE474/574 (Introduction to Machine Learning) or equivalent. A strong background in linear algebra and probability/statistics is expected.

1.4 Programming Requirements

All in-class demonstrations and homework-related codes will be in `Python 3.x`. We will use `GPy`, a native library for Gaussian Process (See https://sheffieldml.github.io/GPy/ for details and installation instructions). Additionally, we will also explore two modern libraries, GPyTorch and GPflow to develop scalable GP based solutions for large scale problems.

1.5 CSE MS Requirements

- The final project in this class satisfies the MS project requirement for the CSE Masters program.
- This class will be considered as an AI focus area course, with regards to the CSE graduate requirements.

1.6 Topic Schedule

Week	Topic	Notebook	Resources
1	Introduction	Primer Introduction	
1	Machine Learning Models	Models	[PONotes] Chapter 1
2	Introduction to Gaussian Processes		[GPMLBook] Chapters 1 and 2
3-4	Gaussian Process Regression		[GPMLBook] Chapters 2 and 4
5-6	Classification with GPs		[GPMLBook] Chapter 3
7	Model Selection for GP Models		[GPMLBook] Chapter 5
8	Scaling GPs for Big Data		[GPMLBook] Chapter 8
9	Handling Complex Scenarios with GPs		Selected readings
10	Dimensionality Reduction with GPs		Selected readings
10	Bayesian Optimization with GPs		Selected readings
11	Deep Learning with GPs		Selected readings
12	Clustering and Dirichlet Process		[PONotes] Chapter 2
13	*Thanksgiving Break*	*No class*	
14	**Final Project Presentations**		

1.7 Course Deliverables

Note:

- Homeworks
  - Will be released every Tuesday at 9.00 AM EST
  - Due in two weeks before the end of the Tuesday lecture
  - Homework 0 will not be evaluated (warm up)
  - All homeworks will be submitted electronically on UBLearns
  - Homeworks will include mathematical derivations, analytical proofs and data analysis

1.8 Term Project

- Students will work in groups of 3 on a semester long project.
- The choice of topic will be flexible but should involve using the models discussed in class.
- A two-page project description, detailing the plan, expected outcomes, and milestones, will be due (submitted electronically via UBLearns) on September 29.
- A two-page mid-semester report project description, detailing the plan, expected outcomes, and milestones, will be due (submitted electronically via UBLearns) on September 29.
- Each group will make a 15 minute project presentation during the last week of classes (Dec 1 and 4) detailing their findings.
- A 5-10 page final project report, detailing the methodology and findings of the project, will be due at the end of the semester on December 15th.

1.9 Course Texts

- [GPMLBook] Carl Rasmussen and Christopher Williams, Gaussian Process for Machine Learning, MIT Press, 2006.
- [PONotes] Peter Orbanz, `Lecture Notes on Bayesian Nonparametrics`, *unpublished*, 2014.

1.10 Grading

- Homeworks (6) – 60%
- Final Project – 40%
- Final grade cut-offs (*TBA*)

1.11 Exams

- This course has no exams

1.12 Expectations

- Students are expected to act in a professional manner. A student's grade may be reduced due to unprofessional or disruptive behavior. Examples include coming to class late, texting (or otherwise using your cell phone) during class, your cell phone ringing during class and/or exams, etc.
- Homeworks will be graded and returned to students.
- Late submission of homeworks will receive a grade of zero.
- Students are encouraged to discuss homeworks and share ideas, but each student must independently write and submit their own solution.

1.13 Accessibility Services and Special Needs

If you have a disability and may require some type of instructional and/or examination accommodation, please inform me early in the semester so that we can coordinate the accommodations you may need. If you have not already done so, please contact the Office of Accessibility Services (formerly the Office of Disability Services) University at Buffalo, 25 Capen Hall, Buffalo, NY 14260-1632; email: stu-accessibility@buffalo.edu Phone: 716-645-2608 (voice); 716-645-2616 (TTY); Fax: 716-645-3116; and on the web at http://www.buffalo.edu/accessibility/. All information and documentation is confidential. The University at Buffalo and the School of Engineering and Applied Sciences are committed to ensuring equal opportunity for persons with special needs to participate in and benefit from all of its programs, services and activities.

1.14 Academic Integrity

This course will operate with a zero-tolerance policy regarding cheating and other forms of academic dishonesty. Any act of academic dishonesty will subject the student to penalty, including the high probability of failure of the course (i.e., assignment of a grade of "F"). It is expected that you will behave in an honorable and respectful way as you learn and share ideas. Therefore, recycled papers, work submitted to other courses, and major assistance in preparation of assignments without identifying and acknowledging such assistance are not acceptable. All work for this course must be original for this course. Additionally, you are not allowed to post course homeworks, exams, solutions, etc., on a public forum. Please be familiar with the University and the School policies regarding plagiarism. Read the Academic Integrity Policy and Procedure for more information: http://undergrad-catalog.buffalo.edu/policies/course/integrity.shtml. Visit the Senior Vice Provost for Academic Affairs web page for the latest information at http://vpue.buffalo.edu/policies/

> **Machine Learning Honor Code**
>
> Against the ML honor code to:
>
> 1. Submit someone else's work, including from the internet, as one's own for any submission
> 2. Misuse Piazza forum
>
> You are allowed to:
>
> 1. Have discussions about homeworks. Every student should submit own homework with names of students in the discussion group explicitly mentioned.
> 2. Collaborate in a group of 3 for the final project. One submission is required for each group.

Warning:

- Violation of ML honor code and departmental policy will result in an automatic F for the concerned submission
- Two violations fail grade in the course

CHAPTER 2

Project Ideas

Here are a few ideas for the course project (to be done in groups of 3). These are very high-level suggestions, and you will have to "fill in the gaps". Feel free to discuss possible directions with the instructor.

Note: If you are currently working on a research project, as part of a different course or towards your MS/PhD thesis, and you would like to use that for the course project, please discuss with the instructor. This will be allowed, as long as a clear project plan is presented along with a discussion on how the course project will be different from the original research project.

2.1 Optimal model selection with Gaussian processes

Choose a problem that you like (connected to your research, or some publicly available data set) and apply GP learning to it. You can choose a regression, classification, or a dimensionality reduction problem. Determine the optimal kernel for the task. How will you model any structural relationships (temporal, spatial, network-based) in the data?

2.2 Scaling Gaussian Processes

One of the key challenges with GP learning is the computational complexity. In this course, we will discuss several strategies to the alleviate that issue. Choose a *big* problem and apply different scaling strategies. What impact does each have on the computational complexity and the accuracy of the model.

2.3 Bayesian optimization using Gaussian Processes

Bayesian optimization is an optimization strategy that can be used when one cannot calculate the gradient of a function, and it has some strong applications in hyper-parameter tuning for machine learning algorithms, including deep neural networks. Central to Bayesian optimization is the use of GP to approximate the objective function. Demonstrate the use of GP for Bayesian optimization, using a real-world problem and a machine learning algorithm, on the task of identifying the best hyper-parameters.

2.4 Deep learning methods and Gaussian Processes

There are connections between deep learning and GPs. However, here the task to understand the benefits and shortcomings of each method in solving a real-world problem. What can you get from GP that you cannot get from neural networks? Think predictive uncertainty and input uncertainty. Can the two be combined?

CHAPTER 3

Computing Resources

All students will access to a GPU-enabled node. To access the resource do the following:

1. Go to the following link from your browser:

```
https://rdweb.wvd.microsoft.com/webclient/index.html
```

2. Use your *<UBIT ID>@buffalo.edu* email address and password to login.

> **Warning:** You will have to go through the DUO two-factor authentication process to login.

3. Choose either *Jupyter Lab* or *Jupyter Notebook* resource from the following page. You will be asked to authenticate again using your UB email and password.
4. You can now create new notebooks and utilize the resource.

> **Note:** While the Python environment has several relevant libraries, such as `numpy`, `pandas`, `matplotlib`, `GPy`, `scikit-learn`, `GPFlow`, `Tensorflow+Keras`, contact me if you need any additional libraries.

5. You can use the **One Drive** resource to upload data files to the environment. The resource is available from your browser by going to:

```
https://onedrive.live.com/about/en-us/signin/
```

6. You will have to go through the same authentication process to get to the drive. You can upload (or download) data here. This folder is accessible in the above cloud environment as the following directory:

```
OneDrive - University at Buffalo
```

3.1 Using TensorFlow

In order to manage the memory of the virtual node, please include the following after importing `tensorflow`

```
import tensorflow as tf
gpus = tf.config.experimental.list_physical_devices('GPU')

if gpus:
   # Restrict TensorFlow to only allocate 1GB of memory to the first GPU
   try:
      tf.config.experimental.set_virtual_device_configuration(gpus[0],[tf.config.
↪experimental.VirtualDeviceConfiguration(memory_limit=711)])
      logical_gpus = tf.config.experimental.list_logical_devices('GPU')
      print(len(gpus),"Physical GPUs,",len(logical_gpus),"Logical GPUS")
   except RunTimeError as e:
      # Virtual devices must be set before GPUs have been initialized
      print(e)
```

CHAPTER 4

Glossary

Term (s)	Description	Notation
Data object (point, observation, sample, example)	A unit of analysis. Typically, a data object is represented as a **vector** of **features**.	Typically denoted as lower case letter, often bold, e.g., $\mathbf{x}$ or $\mathbf{x}_i$ or $\mathbf{x}^{(i)}$, where the subscript or superscript i denotes membership in a **data set**.
Data set	A collection of data objects.	Typically denoted as upper case letter, often bold, e.g., $\mathbf{X}$.
Vector	A list (or array) of real values ($\in \{-\infty, \infty\}$).	Typically denoted as bold lower case letter, e.g., $\mathbf{x}$. $\mathbf{x} \in \mathbb{R}^{\mathbf{d}}$, means that the vector represents a point in a d-dimensional vector space. An individual element of the vector is denoted as x_i.
Matrix	A 2-way array of real values ($\in \{-\infty, \infty\}$).	Typically denoted as bold upper case letter, e.g., $\mathbf{X}$. • $\mathbf{X} \in \mathbb{R}^{\mathbf{m} \times \mathbf{n}}$, means that the matrix has m rows and n columns. A vector is represented as a $m \times 1$ matrix. • An individual element of the matrix is denoted as X_{ij}.
Transpose	$\mathbf{X}^\top$, $\mathbf{x}^\top$	A transpose of a matrix is an operator which flips a matrix over its diagonal, i.e., $X_{ij} = X^\top_{ji}$. Transpose of a vector is a $1 \times d$ matrix.
Matrix multiplication	$\mathbf{X} = \mathbf{YZ}$	Only valid if the number of columns in $\mathbf{Y}$ is equal to the number of rows in $\mathbf{Z}$. • The ij^{th} entry of the matrix $\mathbf{Z}$ is the dot product (see below) between the i^{th} row of $\mathbf{X}$ and j^{th} column of $\mathbf{Y}$, i.e., $Z_{ij} = X^\top_{i*} Y_{*j}$ Where X_{i*} denotes the i^{th} row of $\mathbf{X}$ and Y_{*j} is the j^{th} column of $\mathbf{Y}$
Vector dot (inner) product	$\mathbf{x}.\mathbf{x} = \sum_{i=1}^{d} x_i^2$	In matrix notation, the dot product is expressed as $\mathbf{x}^\top \mathbf{x}$
Data Matrix	A $n \times d$ matrix, $\mathbf{X}$	If each data object in a data set can be represented as a vector in $\mathbb{R}^d$, the data set of n such objects is typically arranged as a $n \times d$ matrix, $\mathbf{X}$, where the transpose of each row of the matrix corresponds to a data object, i.e., $\mathbf{x}_i = X^\top_{i*}$.
Random Variable	A variable whose possible values are outcomes of a random phenomena (distribution)	Typically denoted as an upper case letter, X (bold - $\mathbf{X}$, if multivariate)
Probability	A measure of the	$P(A)$ denotes the probability of an event A to occur.

CHAPTER 5

Other Links

- Piazza
- UBLearns
- iPython Notebooks