Course Syllabus for SIADS 543: Unsupervised Learning Spring/Summer 2021

How to Get Help

If you have questions concerning the degree program, encounter a technical issue with Coursera, or issues using Slack, please submit a report to the ticketing system at umsimadshelp@umich.edu.

If you have an issue specific to the Coursera environment, you can also begin a live chat session with Coursera Technical Support (24/7) or view Coursera troubleshooting guides. (you may be asked to log in to your Coursera account).

For questions regarding course content, refer to the **Communications Expectations** section below.

Course Overview

Unsupervised learning algorithms are methods for transforming and finding structure in datasets without the benefit of labeled examples to guide them. Students will learn how to correctly apply, interpret results, and iteratively refine and tune unsupervised machine learning models to solve a diverse set of problems on real-world datasets. Application is emphasized over theoretical content. The unsupervised learning course is an essential part of the core MADS machine learning series: its concepts, algorithms, and evaluation methods are used heavily throughout the following machine learning courses that include: deep learning and machine learning pipelines.

Prerequisites

Knowledge of key concepts and methods covered in the SIADS 542 Supervised Learning course, as well as familiarity with the scikit-learn, numpy, pandas, and scipy libraries.

Instructor and Course Assistance

Instructor: Yumou Wei (yumouwei@umich.edu)

Course Assistants: Gregory Myers

Course Communication Expectations

Slack is the preferred communication tool for this course. If you have questions about course content (e.g. lecture videos or assignments), please make sure to use Slack. Instructor and course assistant response time to Slack messages will aim to be within 24 hours, Monday-Friday.

Please try to monitor the Slack channels for the course regularly.

Personal communication that may involve sensitive information may be emailed directly to the instructor or course assistant. If you email the instructor or course assistant, please include SIADS543 in the email subject. Instructor and course assistant response time to email messages will aim to be within 24 hours.

Office Hours are held on:

- Mondays 11am Eastern Time (Yumou Wei)
- Thursdays at 10 am Eastern Time (Gregory Myers)

Office hour sessions will be recorded for the benefit of students who are unable to join at these times. Password to join any Office hours is **543**

Technology Requirements

The course programming will be based on Jupyter notebooks and Python 3.x.

Required Textbook

This course will use the following textbook as a reference and source of examples: **Introduction to Machine Learning with Python**, by Andreas C. Müller and Sarah Guido (O'Reilly Media)
This text is available free online via the University of Michigan Library:

1. On the Welcome! screen, choose "Select your institution" to open the menu and select the first option "Not listed? Click here."
2. In the Academic email box, enter your U-M email address (in the format: uniqname@umich.edu).

Users can also create an individual account using your U-M email, but don't have to. There is a more detailed description of access options here. (Unfortunately, to add to this there have been some users recently who have reported error messages when trying to login to this database. My general advice for this problem is to try using an incognito browser window and follow the steps above.)

This text is also available for purchase on the O'Reilly website.

Other Textbooks and Resources (Optional)

From time to time I may refer to examples or other content from the classic textbook **The Elements of Statistical Learning** (Second Ed.) by Trevor Hastie, Robert Tibshirani, and Jerome Friedman, published by Springer.

The entire textbook is free and available for online downloading.

For very useful mathematical background, see the companion webpage to the book "Mathematics for Machine Learning". Copyright 2020 by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Published by Cambridge University Press.

Learning Outcomes

Here's a summary of some key learning objectives we have (1) for the course overall, and (2) broken down by week.

Course-wide objectives

- Correctly apply and interpret results from clustering methods in scikit-learn, including k-means, agglomerative clustering, hierarchical clustering, and DBSCAN.
- Understand the use of topic modeling (Latent Dirichlet Allocation and Non-Negative Matrix Factorization forms) and best practices for its application.
- Correctly apply and interpret results from manifold learning methods, including multidimensional scaling (MDS) and t-SNE.
- Understand how to evaluate clustering results using a variety of metrics.
- Understand the tradeoffs and assumptions inherent in different clustering techniques.
- Understand how unsupervised learning can be used to improve supervised prediction.
- Perform density estimation using a kernel, with a single random variable.
- Interpret a biplot result from principal components analysis (PCA).
- Build awareness of the basic mechanism and use of word embeddings (in preparation for later coverage in deep learning).
- Build awareness of the EM algorithm: what it does, how and why it's used, and how it relates to clustering.
- Build awareness of other advanced methods like kernel PCA and spectral clustering.

Week 1

- Apply PCA to a dataset: create and interpret biplot.
- Understand the Singular Value Decomposition.
- Apply MDS and t-SNE to a dataset, interpret results.
- Learn how normalization should be applied to input, and how key parameters can affect output.
- Perform density estimation on a single variable using different kernel choices/parameters.

Week 2

- Apply k-means clustering to a given dataset.
- Learn about issues with applying some clustering methods in practice, such as local minima and restarts.

- Create a dendrogram from hierarchical data to answer questions about the dataset.
- Use DBSCAN to find groups and detect outliers.
- Compare different clusters in terms of selected quality metrics.

Week 3

- Learn about the Expectation-Maximization (EM) algorithm.
- Apply LDA and NMF topic modelling to a text dataset, compare results.
- Learn about input text representations (e.g. tf.idf) and how this can affect results.
- Understand Latent Semantic Indexing and how it can be used for semantics-based text matching.
- Use of word2vec embedding for text similarity (compared to simple word overlap).

Week 4

- Apply unsupervised learning techniques to a real-world dataset, including the methods introduced this week.
- Use unsupervised methods to find features for a supervised learning problem.
- Learn how unsupervised learning can be used for data imputation.
- Learn about related methods: self-supervised learning and semi-supervised learning.

Schedule

Please note: *in order to provide a self-contained module for some topics (e.g. where it's easy to move back and forth for reference if needed), a few videos turned out to be significantly longer than average. Also, in contrast to supervised learning, the unsupervised learning course covers more material in the first week than the supervised learning course did, so please plan your schedule accordingly. Unlike in supervised learning where the first assignment was shorter and given less weight, all four assignments in unsupervised learning have *equal* weight.*

Week 1: You'll be introduced to basic unsupervised learning methods that focus on transformation of data: dimensionality reduction, manifold learning, and density estimation, with analysis of realistic datasets, implemented using the scikit-learn library. For this week's assignment you'll apply Principal Components Analysis to gain insight into a large real-world dataset, use manifold learning methods such as t-SNE to visualize complex structure, and use kernel density estimation to estimate probabilities of conditional events.

Week 2: This week will focus entirely on clustering - another critical and widely-used unsupervised learning method. You'll learn about the most important families of clustering algorithms: hierarchical methods (agglomerative bottom-up, divisive top-down), partitioning methods (k-means, k-medoids) and density-based methods (DBSCAN). You'll also gain awareness of more advanced methods such as spectral clustering, and how to evaluate cluster

quality. This week's assignment will have you apply a variety of these clustering approaches to realistic datasets using scikit-learn's clustering capabilities.

Week 3: Our theme this week is estimating latent variables, another important area of unsupervised learning, especially for text-based applications. We'll cover the EM algorithm for estimating latent variables and its connection with k-means clustering. Topic modeling is another form of latent variable estimation and you'll learn about two different methods for this: Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing. We'll also survey word embeddings: learning how to represent words with vectors in semantically useful ways. This week's assignment will include problems that have you apply EM to a new scenario, analyze topic structure in a large document collection, and apply word embeddings to an NLP-related task.

Week 4: In the final week of this course, we'll see how unsupervised methods can be integrated with supervised learning methods to improve prediction performance. To do this, we'll look at various special topics, including data imputation (dealing with missing data) and extensions of unsupervised learning that are at the cutting edge of today's technology: semi-supervised learning and self-supervised learning. This week's assignment will be a synthesis project in which you apply unsupervised methods and supervised methods to a complex real-world dataset.

Assignments

Week 1: Apply PCA to a dataset: create and interpret biplot. Apply MDS and t-SNE to a dataset, interpret results. Learn how normalization should be applied to input, and how key parameters can affect output. Perform density estimation on a single variable.

Week 2: Apply k-means clustering to a given dataset. Learn about local minima and restarts. Create a dendrogram from hierarchical data to answer questions about the dataset. Compare different clusters in terms of selected quality metrics.

Week 3: Apply LDA and NMF topic modelling to a text dataset, compare results. Learn about how the input text representation (e.g. tf.idf) can affect results. Use of word2vec embedding for text similarity (compared to simple word overlap).

Week 4: Apply various unsupervised learning techniques covered as special topics, to real-world data, including using supervised methods to help solve specific unsupervised learning problems like data imputation.

Quizzes

Each week will also contain a short quiz to test your knowledge of material in the lectures and readings.

Grading and Course Checklist

I anticipate no major changes to this course grading scheme. However, as the course progresses, I reserve the right to offer additional bonus assessments or make minor adjustments/fixes as required, for any evaluation method in this course. If necessary, any such changes will always be done in a way that maximizes a student's grade across options.

You must complete all assignments and quizzes to get credit for this course.

Course Assignment	**Percentage of Final Grade**	**Passing Threshold**
Week 1 Quiz	5%	80%
Week 1 Jupyter Notebook Assignment	20%	
Week 2 Quiz	5%	80%
Week 2 Jupyter Notebook Assignment	20%	
Week 3 Quiz	5%	80%
Week 3 Jupyter Notebook Assignment	20%	
Week 4 Quiz	5%	80%
Week 4 Jupyter Notebook Assignment	20%	
Total	**100%**	

Late Submission Policy

Important! Please read and understand this section, and if anything is unclear, it is your responsibility to contact the instructors so that you understand the policy. We realize that, now more than ever, the occasional crisis might mess up your schedule enough to require a bit

of extra time in completing a course assignment. Thus, we have instituted the following flexible late policy that gives you a limited number of flexible "late day" credits.

You have a total of two (2) free late days to “spend” during the course. One late day equals exactly one 24-hour period after the due date of the assignment (including weekends). No fractional late days: they are all or nothing.

As an example, suppose you had two course late days left. Submitting one specific assessment (quiz or assignment) any time within 24hrs of the original due date counts as using the first late day for that assessment. Beyond that time, submitting any time within the next 24h counts as using the 2nd late day for that assessment. After that, each additional 24h period accrues a 15% per day penalty as follows:
Once you have used up your late days, there is a 15% penalty for each subsequent 24-hour period after the deadline that an assignment is late. For example, if the due date is 11:59pm Tuesday, and you have *no* late days left, penalties would be:
Submit before 11:59pm Tuesday: 15% deduction
Submit before 11:59pm Wednesday: 30% deduction
Submit before 11:59pm Thursday: 45% deduction
Submit after 11:59pm Thursday: 60% deduction

You don't need to explain or get permission to use late days: we will track them for you. We will allocate any late days you have used at the end of the course, after all quizzes and assignments are submitted, so that we can do the allocation in a way that maximizes your final grade. Note that resubmissions after the deadline will be counted as late submissions.

This flexible system is difficult/impossible to implement in Coursera, so basically you may see temporary penalties in the Coursera gradebook but at the end of the course we will add back the late day credits to your grades. We wait until the end to do this so that we can see all your submitted assessments and allocate the late days in a way that maximizes your final grade.

Please note: Submitting your work on time is very important in this course. The instructional team may periodically reach out to you and ask you about your progress; if you fall behind it may be difficult to catch up, and you will be at risk for not succeeding in the course.

Letter Grades

The grading scale for this course will be as follows:

A+	97%
A	93%
A-	90%

B+	87%
B	83%
B-	80%
C+	77%
C	73%
C-	70%
D+	67%
D	63%
D-	60%
F	0%

Program-wide Information

Help Desk(s): How to get Help

Need help? You may reach out to UMSI or Coursera depending on the type of question you have.

- Degree program questions or general help - umsimadshelp@umich.edu
- Coursera's Technical Support (24/7) - https://learner.coursera.help/

Academic Integrity/Code of Conduct

Refer to the Academic and Professional Integrity section of the UMSI Student Handbook. (access to Student Orientation course required).

Accommodations

Refer to the Accommodations for Students with Disabilities section of the UMSI Student Handbook (access to the Student Orientation course required). Use the Student Intake Form (requires U-M login) to begin the process of working with the University's Office of Services for Students with Disabilities.

Accessibility

Refer to the Screen reader configuration for Jupyter Notebook Content document to learn accessibility tips for Jupyter Notebooks.

Library Access

Refer to the U-M Library's information sheet on accessing library resources from off-campus. For more information regarding library support services, please refer to the U-M Library Resources section of the UMSI Student Handbook (access to the Student Orientation course required).

Student Mental Health

Refer to the University's Resources for Stress and Mental Health website for a listing of resources for students.

Student Services

Refer to the Introduction to UMSI Student Life section of the UMSI Student Handbook (access to the Student Orientation course required).

Technology Tips

We will be using Slack, Zoom, Google Docs, and Google Sheets to facilitate communication. Your own work on the project will be done in Jupyter.

We have created a Jupyter environment for you that is functionally equivalent to SIADS 516, which is a superset of the base MADS environment. You can access that environment via the "ungraded lab assignment" in Coursera. You can use that environment or choose to use any of the environments from courses you have already completed. Alternatively, you can use your own locally installed environment. Another possibility is to use Google Colaboratory, which may facilitate collaboration.

Working Offline

While the Coursera platform has an integrated Jupyter Notebook system, you can work offline on your own computer by installing Python 3.5+ and the Jupyter software packages. For more details, consult the Jupyter Notebook FAQ.