

Machine Learning for Science & Society

Contents

Syllabus

- [Basic Facts](#)
- [Schedule](#)
- [Learning Outcomes and Evaluation](#)

Notes

- [Spring 2021](#)
- [Spring 2022](#)

Reference

- [Preparing to present a paper](#)
- [Posting Notes](#)

- Spring 2022
- time: MW 3:00-4:15pm
- Professor: [Sarah Brown](#)
- course number: CSC 592: Topics in Computer Science
- Credits: 4
- Location: Tyler Hall 108

In this class, we will address the challenges in applying machine learning to scientific research and in high stakes social contexts. On the science side, we will examine the role of ML in research, in particular how it works within knowledge production and how to evaluate ML in line with domain norms. On the social side, we will consider how to ensure ML-based algorithmic decision making systems uphold social values, with a focus on fairness. While these two applications are distinct, many of the challenges translate into common technical problems. Some of the common challenges include:

- missing data
- noisy or missing labels
- multiple objectives

We will look at a range of strategies for identifying and mitigating these problems including:

- robust evaluation
- model inspection
- explanations
- interpretable models

Format

This will be a synchronous course offered in person.

The course will involve:

- reading and evaluating ML research papers
- facilitating and participating in class discussions of the papers
- producing a replication, demo, or illustration of one concept covered for a broader audience

- completing a project using ML in a scientific or social domain
- writing a CS conference style (short & concise) final paper on their project

graduate students are encouraged to do a project related to their research

Prerequisites

To be successful in this class students should have:

- past experience with machine
- basic programming skills
- familiarity with concepts in probability, linear algebra, and calculus that appear in ML

varying skill in these topics is ok, but a general understanding of the basic ideas is important.

[Complete this Google form](#) to request a permission number from Professor Brown to enroll in this course. Note that you must be enrolled at URI to take this course and be logged into your URI google account to view that form.

Basic Facts

Meetings

This class will meet on Monday and Wednesday 3-4:15pm in person.

Instructor

Professor Sarah M Brown is an Assistant Professor in Computer Science. Her current research aims to answer the question, "How can machine learning produce AI systems that make fair decisions?"

Office Hours

By appointment, link on Brightspace.

Schedule

We meet in Tyler 108, MW 3-4:15pm.

This course will proceed in three main parts: overview, deep dives, and wrap up.

Structure

Overview

In the first part of the course we will review ML basics, set norms for interaction and complete a survey of the topics that we will cover for the rest of the semester.

In this part of the class, Professor Brown will lead synchronous sessions. Students will be responsible for reading overviews, refreshing background material, and choosing an area for their course project. Students will start with an introductory demo or replication as a mini project.

Deep Dives

During the middle of the course we will spend one week on each topic. There will be 1-3 papers to read each week.

Students will be responsible for presenting papers in class on a rotating basis.

During this time students will have milestones where they need to complete interim steps for their course project. The first milestone will be a proposal that includes the specific products for the remainder of the milestones based on a template.

Conclusion

We will also workshop students' projects, giving substantive feedback prior to the final submissions.

Final projects will be evaluated through a presentation and paper

Weekly topics

The readings are subject to revision in class up until a presenter is assigned. Topics may also be updated after the first few classes based on student interests and recent publications.

Class	Topic	Reading	Activities
2022-01-24	Introduction	None	introductions, expectation setting
2022-01-26	Probability Review	Model Based ML, chapter 1	reading discussion, setting up
2022-01-31	Setting the Stage	The Scientific Method in the Science of Machine Learning and Value-laden Disciplinary Shifts in Machine Learning	Paper Presentation by Dr. Brown
2022-02-02	Meta issues	Roles for computing in social change	Paper Presentation by Dr. Brown
2022-02-07	Missing Data: Intro strategies	Handling Missing Values when Applying Classification Models & Missing data imputation using statistical and machine learning methods in a real breast cancer problem	Paper discussion led by Emmely & Chan
2022-02-09	Missing data with graphical models and causal reasoning	Graphical Models for Inference with Missing Data	Paper discussion led by Chamudi
2022-02-14	causal and probabilistic missing data	Missing Data as a Causal and Probabilistic Problem	Paper discussion by Lily
2022-02-16	Fairness	Fairml classification chapter and Machine Bias and Gender Shades and Obermeyer	Paper discussion by Damon & Dereck
2022-02-23	Fairness and Causality	FairML Causality chapter Empirical comparison paper	Paper discussion Surbhi and Chan
2022-02-28	Multi-objective & constrained opt	Elastic Net	Paper presentation by Alex
2022-03-02	Multi-objective & constrained opt	A critical review of multi-objective optimization in data mining: a position paper	Paper presentation and discussion by Chamudi
2022-03-07	Latent Variable Models	Gaussian Mixture Models and Topic Models	Paper presentation by Alex and Lily
2022-03-09	Latent Variable Models	Indian Buffet Process and Auto-Encoding Variational Bayes	Paper presentation by Surbhi and Chan
2022-03-28	Missing or Noisy labels	Learning with Noisy Labels and Semi Supervised Learning	presentations by Dereck
2022-03-30	Noisy Labels as a model for Bias	Recovering from biased data: Can fairness constraints improve accuracy and Fair classification with group dependent label noise	Paper presentations by Surbhi and Lily
2022-04-04	Interpretable & Explanation Intro	A Survey of Methods for Explaining Black Box Models	Paper Presentation by Emmely
2022-04-06	A Case for Interpretability over Explanation	Why are we explaining black box models and Learning Certifiably optimal rule lists for categorical data	Paper Presentation by Alex and Surbhi

Class	Topic	Reading	Activities
2022-04-13	Models for Explanation	Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)	Paper Presentation Dereck
2022-04-18	Choosing Explanations and using explanations	How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations Actionable Recourse in Linear Classification	Paper Presentation by Alex and Chan
2022-04-20	What does Interpretable mean	Towards A Rigorous Science of Interpretable Machine Learning and Towards falsifiable interpretability research	Paper Presentation by Sarah
2022-04-25	3 Project Presentations	projects	Paper Presentation by Alex, Dereck, and Chamudi
2022-04-27	3 Project Presentations	projects	presentations by Emmely and Lily
2022-05-02	2 Project Presentations Chan and Surbhi	projects	presentations with peer feedback; discuss revision plans; overall course review

Table 1 Schedule

Learning Outcomes and Evaluation

Important

This will change for spring 2023

This course has goals with respect to the knowledge and research skills.

Evaluation will be with respect to each of the outcomes and based on a level of mastery: general awareness, competency, or mastery.

By the end of the course students will be able to:

- Critique common ways that social or scientific applications of ML require violating ML algorithm assumptions and ways to mitigate or adapt.
- Evaluate ML Research papers for their applicability to scientific and social applications of ML.
- Communicate about ML and its limitations work to varied audiences
- Apply ML to scientific and social data responsibly

Activities

- reading and evaluating ML research papers
- facilitating and participating in class discussions of the papers
- coproducing notes that summarized key points and open questions of papers
- producing a replication, demo, or illustration of one concept covered for a broader audience
- completing a project using ML in a scientific or social domain
- writing a CS conference style (short & concise) final paper on the project

Evaluation

The grading scheme is rooted in achieving the learning outcomes and finalized with a grading contract. Each student will submit a grading contract in the first two weeks and then if all work meets the specification, will earn the contracted grade.

The following describes each activity in the course and the specification for it.

Discussions, Exercises, and Notes

For each topic we cover in class, you should engage fully in the class discussion and practice exercises that are provided if applicable.

To demonstrate engagement you must:

- provide a good faith attempt at any exercises provided
- contribute to the discussion (comments and questions both count)
- contribute to annotated class notes

Presentations

Presenting papers and participating in class will contribute to demonstrating a basic awareness at each of learning objective.

Each class session will be evaluated on if you contribute to discussion or not. This includes both asking questions and answering questions.

Each time you present will be evaluated on specification, your presentation should:

- summarize the key takeaways for the reading(s) in your own words
- summarize key details for understanding to facilitate the discussion
- discussion of strengths and weaknesses of the paper & method
- describe how this paper relates to bigger ideas in the course or your own work

You'll present 2-3 times and you will be expected to improve each time, not to be perfect.

When you present you don't have to have all the answers, you can have open questions.

The goal is that you guide the discussion by doing the above and opening the floor up for questions.

Project

The final project is a chance to dive deeply into one of the course topics. It has the following timeline. Percentages below are of the total grade.

Date	Milestone	Submission format	Evaluation
2022-02-18	Area Selection	Consultation meeting and general questions	feedback only
2021-03-02	Topic Selection	Objectives and scope of work	completion or scope adjustment
2021-03-10	Proposal	Problem statement, lit review, method	specification, with revisions
2021-04-02	Checkin	Consultation meeting and prelim result	completion
2021-04-20	Rough draft	Draft ready for peers to read	feedback only, per paper specs
2021-04-25+	Presentation	talk in class	specification
2021-05-02*	revision plan	plan for final revision, minor extensions	feedback only, per paper specs
2021-05-09	final paper	final paper submitted for grading	specification
2021-05-09	final reflection	final paper submitted for grading	completion

+presentation dates are on the [schedule](#) if presenting on 5/2, revision plan is due 5/4

Proposal Specifications

Submit a 1.5- 2page proposal in the ACM Proceedings format.

Your proposal should include a concise problem statement, a preliminary literature review that situates your project, a description of method(s) you will use to answer your questions in your project, and the expected outcomes of your project.

The proposal will be graded on if it meets the specification or not, but you will be able to revise and resubmit if the first submission does not. To meet specification it must:

- be the right length
- be the right format
- include all sections
- be written clearly
- describe the problem, clearly identifying what the specific goals of your project are
- describe a tractable project
- summarize relevant literature for the problem context
- summarize relevant course-related literature for your project
- describe what you will do in your project
- describe what the end outcome of your project.

Presentation Specifications

Your presentation should:

- include an agenda for the talk
- describe the problem
- summarize relevant background
- clearly identify what you did
- describe findings
- include concluding remarks on reflection/possible extensions
- be 15-20 minutes long

there will be a 5-10 minute Q&A after

Paper Specifications

Your final paper should include a concise problem statement, a complete literature review that situates your project, a description of method(s) used your project, findings, and a discussion or future work section.

For it to meet specification it must:

- be the 4-5 pages + references
- be the right format
- include clearly marked sections indicating the required content
- be written clearly
- describe the problem, clearly identifying the specific goals of your project
- summarize relevant literature for the problem context
- summarize relevant course-related literature for your project
- include clear description of what was accomplished
- include a clear summary of results (may include null results/ failed findings)
- include a conclusion that suggests possible next directions

Reflection

Add to your grading contract repo:

- summarize what you learned in your project

- any challenges and how you over came them/ or wish you had asked for more help

Translation Mini Project

For this assignment you can choose any topic other than the one your project is for and produce a short demo, illustration, or replication that makes some aspect of the the topic accessible for a broader audience.

For this, you must submit a one paragraph proposal that describes your demo Once that's approved that it will count, you have two weeks to build your demo or replication. The latest your demo may be submitted is at the same time as your final project.

The proposal will be graded on specification and may be resubmitted until successful. Your demo proposal must:

- state the topic from class your demo relates to
- state the format/medium your demo will take:
 - illustration, replication, interactive visualization, etc
- describe the target audience (a particular type of scientists, impacted people, software engineers, layperson, etc)
- describes what your demo will do by answering the relevant questions from the list below:
 - what will a person learn by reading/ using your demo?
 - if it's interactive what will vary? what will be the inputs?
 - what specific result will you replicate?
- describe a demo that is an appropriate scope (not too large or too small)

The demo will be graded on specification and can be revised and resubmitted one time. Your demo must:

- describe a topic accurately
- be accessible to the specified topic model
- meet the description in the proposal

With your demo or after, submit a one paragraph reflection describing what you learned doing this exercise. The reflection will be graded on completion.

Spring 2021

Schedule

We will meet synchronously via Zoom: Tu 5:30-8:15

This course will proceed in three main parts: overview, deep dives, and wrap up.

Structure

Overview

In the first part of the course we will review ML basics, set norms for interaction and complete a survey of the topics that we will cover for the rest of the semester.

In this part of the class, Professor Brown will lead synchronous sessions. Students will be responsible for reading overviews, refreshing background material, and choosing an area for their course project. Students will start with an introductory demo or replication as a mini project.

Deep Dives

During the middle of the course we will spend one week on each topic. There will be 1-3 papers to read each week.

Students will be responsible for presenting papers in class on a rotating basis.

During this time students will have milestones where they need to complete interim steps for their course project. The first milestone will be a proposal that includes the specific products for the remainder of the milestones based on a template.

Conclusion

In the end of the course, we will focus on integrating ideas across multiple topics.

We will also workshop students' projects, giving substantive feedback prior to the final submissions.

Final projects will be evaluated through a presentation and paper

Weekly topics

Class	Topic	Reading	Activities
2021-01-29	Introduction	None	introductions, expectation setting
2021-02-01	Probability Review	Model Based ML, chapter 1	reading discussion, setting
2021-02-03	ML Process & Mutual information preview	Scikit learn getting started,	live coding
2021-02-08	Missing Data: Intro strategies	Handling Missing Values when Applying Classification Models & Missing data imputation using statistical and machine learning methods in a real breast cancer problem	Paper discussion led by Daniel
2021-02-10	Missing data with graphical models and causal reasoning	Graphical Models for Inference with Missing Data & Missing Data as a Causal and Probabilistic Problem	Paper discussion led by Julian
2021-02-15	Current Challenges in Missing data	Handling Missing Data in Decision Trees: A Probabilistic Approach & How to miss data? Reinforcement learning for environments with high observation cost	Paper discussions by Xavier and Zhen
2021-02-17	Current Challenges in Missing data	How to deal with missing data in supervised deep learning	Paper discussion by Madhukara, Replication & testing discussion,
2021-02-22	Fairness	fairml classification chapter and friedler empirical comparison paper	Empirical setup
2021-02-24	Fairness	Reading	preview of lasso and admm constraint to multiobjective reformulation
2021-03-01	Multi-objective & constrained opt	Elastic Net	Paper presentation by Daniel, try out elastic net & LASSO in scikit learn
2021-03-03	Multi-objective & constrained opt	A critical review of multi-objective optimization in data mining: a position paper	Paper presentation and discussion by Zhen
2021-03-08	Latent Variable Models	Gaussian Mixture Models and Topic Models	Paper presentation by Xavier
2021-03-10	Latent Variable Models	Indian Buffet Process and Auto-Encoding Variational Bayes	Paper presentation by Madhukara
2021-03-15	Missing or Noisy labels	Learning with Noisy Labels and Semi Supervised Learning	Julian and Daniel
2021-03-17	Noisy Labels as a model for Bias	Recovering from biased data: Can fairness constraints improve accuracy and Fair classification with group dependent label noise	Zhen
2021-03-22	Interpretable & Explanation Intro	A Survey of Methods for Explaining Black Box Models	Xavier

Class	Topic	Reading	Activities
2021-03-24	A Case for Interpretability over Explanation	Why are we explaining black box models and Learning Certifiably optimal rule lists for categorical data	Madhukara
2021-03-29	Models for Explanation	Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV) and A unified approach to interpreting model predictions	Zhen
2021-03-31	Choosing Explanations and using explanations	How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations Actionable Recourse in Linear Classification	Daniel
2021-04-05	What are the risks of explanations	Model Reconstruction from Model Explanations	Xavier
2021-04-07	What does Interpretable mean	Towards A Rigorous Science of Interpretable Machine Learning and Towards falsifiable interpretability research	Madhukara
2021-04-12	Meta issues	The Scientific Method in the Science of Machine Learning and Value-laden Disciplinary Shifts in Machine Learning	Sarah
2021-04-13	Meta issues	Roles for computing in social change	Sarah
2021-04-19	Project Presentations	projects	presentations with peer feedback
2021-04-21	Project Presentations	projects	peer feedback
2021-04-26	Review and Project Reflections	Paper feedback	presentations with revision plans

Table 2 Schedule

Class 1: Introductions

Introductions & Goals

Course Admin

- Brightspace
- Zoom
- Google docs or markdown in the future?
- Website

Learning outcomes

knowledge research

- identify common problems and solutions in scientific application of ML
- identify common challenges and solutions for social applications: fairness,

- implement and extend research papers

Activities

- reading and evaluating ML research papers
- facilitating and participating in class discussions of the papers
- producing a replication, demo, or illustration of one concept covered for a broader audience
- completing a project using ML in a scientific or social domain
- reflect on methodologies used in this type of research
- writing a CS conference style (short & concise) final paper on their project

Model Based ML and this course

<https://www.mbmlbook.com/toc.html>

- missing data
- noisy or missing labels
- multiple objectives

We will look at a range of strategies for identifying and mitigating these problems including:

- robust evaluation
- model inspection
- explanations
- interpretable models

ML and Probability Review

admin

- collaborative notes
- brightspace will be updated later this week
- grading details by Wed
- environment for coding demos

More formalism

- model
- prediction algo
- cost function
- objective

Probability

- sample distros

Practical Application of ML & Pipelines

Class 3: ML Pipelines

Goals when using ML

1. Understand about the data (data science/ actual science) probability more statistics, maybe fit another examine model parameters, inspect them
2. understanding about Naive bayes fit different data varies
3. claims about the learning algorithm run multiple algorithms on the same data possibly multiple data

Basic setup

1. test train
2. training parameters

3. estimator objects
4. fit model parameters
5. metrics
6. cross validation

```
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report
from sklearn import datasets
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
/tmp/ipykernel_1684/2975186671.py in <module>
      2 import seaborn as sns
      3 import numpy as np
----> 4 from sklearn.model_selection import cross_val_score
      5 from sklearn.model_selection import train_test_split
      6 from sklearn.naive_bayes import GaussianNB

ModuleNotFoundError: No module named 'sklearn'
```

```
iris_df = sns.load_dataset('iris')
```

```
sns.pairplot(iris_df, hue='species')
```

```
X, y = datasets.load_iris(return_X_y=True)
```

```
X.shape
```

```
y.shape
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
```

```
gnb = GaussianNB()
```

```
gnb.__dict__
```

```
gnb.fit(X_train, y_train)
```

```
gnb.__dict__
```

```
X_test[0]
```

```
y_pred = gnb.predict(X_test)
```

```
y_pred[:5]
```

```
y_test[:5]
```

```
confusion_matrix(y_test, y_pred)
```

```
gnb.score(X_test, y_test)
```

```
gnb2 = GaussianNB(priors=[.5, .25, .25])
```

```
gnb2_cv_scores = cross_val_score(gnb2, X_train, y_train)
```

```
np.mean(gnb2_cv_scores)
```

```
gnb_cv_scores = cross_val_score(gnb,X_train,y_train)
```

```
np.mean(gnb_cv_scores)
```

```
print(classification_report(y_test,y_pred))
```

```
gnb.predict_proba(X_test)
```

Class 4: Missing Data: Basic techniques

Evaluation of missing data at training

- multiple imputation
- ML based was better than imputation which is better than dropping samples
- example datasets: 45% of patients have at least 1 missing value

Imputation

- Mean imputation:
 - insert the mean based on the other values
- Hot deck
 - mean-like with similarity
- Multiple imputation
 - 3 diff ways

Imputation ML

- MLP
 - fully connected
- Self organization
 - competitive learning
 - NN on module of nodes in 2d grid,
- KNN
 - select closest complete case to impute values from
 - expensive for large datasets due to need to search everywhere for each missing value

Testing

- Train NN based on data imputed with each technique

Conclusions:

- in general, any imputation was better than deletion
- ML based performed better

Discussion & Questions

- interesting that even simple methods provide improvement
- SOM is sort of unclear how does that work?
- Review of MLP and [sigmoid](#)

Handling missing values At application time

- reduced models vs imputation.
- broad approach
- 15 common datasets

Techniques:

- Discard

- Acquire missing values
- Imputation
 - predictive value imputation
 - distribution based
 - unique values
- Reduced Feature Models
 - retrain for different feature models

Feature imputability impacts the distribution or predictive type of imputation

More complex model

- decision tree with bagging
- again, reduced model is the best strategy

Hybrid Models for efficient prediction

- reduced models
- a hybrid is a complete model with stored subset for most common missing features
- Reduced feature ensemble
 - N models for N features
 - each one is missing one feature
 - average these together for final prediction
 - substantial reduction in when there is a single feature is missing
 - combine with imputation for multiple features
 - relative accuracy is better than imputation

General takeaways

- reduced models vs imputation is a large improvement
- this is sort of an imputation

Weaknesses

- Didn't check unique value imputation
- MCAR
- focused on

Overall Discussion

- How might the two problems interact?
 - if missing data at both train and prediction...
 - train using missing data without imputation for training the separate models
- Questions on these ideas
- What additional things might you need to consider when choosing one?
 - feature imputability at training
- what to do with time series data
- How to check if missing CAR?
 - look at collection technique
 -
- what do to with varying data per person
 - LSTM for time series data
 - hierarchichal modeling other wise
 - [example of hierarchical with time series also](#)

For Wednesday

1. [Graphical Models for Inference with Missing Data](#)
2. [Missing Data as a Causal and Probabilistic Problem](#)

Missing Data 2:

Graph theory foundation

- A DAG
- shapes are nodes
 - nodes generally represent a random variable
- nodes are connected with edges
 - edges may be directed (with an arrow)
- path is a sequence of edges
- a cycle is a path that returns to a given node twice
- we will focus on acyclic graphs
- directed edges connect parent nodes to child nodes (follow the arrow)
- Why graphs: useful representation of joint distributions
- d-connected: two nodes are d-connected if there is a connected path without a collider
- d-separation: independent through a collider
- collider is when arrows flip

Missingness graphs

- x, y are variables
- Y^* is a proxy for y
- R_y : causal mechanism for missingness of y^*

Recoverability for MCAR

Discussion

- proxy
- example with ocean data temp sensor, cloud cover images

For next week

Choose one: <https://artemiss-workshop.github.io/#program>

Information Theoretic Approaches for Testing Missingness in Predictive Models

<https://openreview.net/forum?id=6Y05VJfGIFM>

Missing Data 3

Handling Missing Data in Decision: A probabilistic approach

key ideas

- A decision tree's structure and notation
- Review of imputation
 - Predictive value imputation
 - mean, median or mode
 - make assumption that features are independent
 - surrogate splits, partition data using another feature to
- XG Boost

Expected Predictions:

- impute all possible completions as once to avoid strong dist assumptions
- consistent for MCAR and MAR
- expensive, but density can help reduce
- tractably compute the exact expected predictions
- loss minimization

Experiments

- for a single dataset, outperforms in general

Discussion

- generally easier
- given single dataset, of results, how much do we trust this?
- what does this provide as an advantage
- NP hard

How to miss data?: Reinforcement learning for environments with high observation cost

Key points

Reinforcement learning

- cost associated with making accurate observations
- goal directed
- RL agent tries to

Problem setting:

- $o \sim p_0(o_t | s_t; \beta)$
- beta is accuracy of obs
- r is old reward

Scenario A:

- observed angle vs

Big picture: manipulating how the data collection

Discussion

- survivorship bias?
- right left imbalance for figure 3
- simple pendulum example helped overcome the background lacking
- figures

General

Try writing out a missingness graph for a problem of choice, some scenario where you imagine there would be missing data, or an example dataset that you can find.

Missing data

supervised

Background

- Hadamard
-

Readings for next week:

http://sorelle.friedler.net/papers/fairness_comparison_fat19.pdf <https://fairmlbook.org/>

- introduction and classification chapters (1 and 2)

Elastic Net

1. OLS can overfit
 2. ridge helps with over fitting, but not variable selection
 3. lasso helps reduce the dimensionality
- $p > n$ lasso saturates at n variables
 - lasso predicts 1 of correlated variables at random
 - ridge is better in correlated case

Multiobjective

Latent Variable Models: GMM & Topic Models

Gaussian Mixture models

- key points:
 - model versus algorithm
- algorithm:
 - initialize
 - Estep
 - Mstep
 - until convergence:
 - parameters stop changing, assignments stop changing
- Covariance types:
 - covers weakness in kmeans

Topic Models

- corpora: collection of documents
- text modeling, was classically binary matrices
 - also tf-idf
 - useful for discriminating documents,
 - lacks meaning
- pLSI: probabilistic, latent semantic indexing
 - reminiscent of GMM
 - assumes exchangeability
 - mixture components

Latent Variable Models:

Semi-supervised learning and noisy labels

Key questions:

How do these relate?

Noisy labels as a Bias model

Fairness constraints for recovering from biased labels

Blum & Stengl

Considers 3 cases:

- more errors in the disadvantaged group than the advantaged group
- fewer positive examples of the disadvantaged group
- both

Comparison of Fairness Interventions

Paper discussion

Spring 2022

Notes will be added after the semester starts.

Overview

Course Info

- graduate course, focused on research adjacent skills
- topic is how to use ML safely and reliably in the context of scientific discovery or social applications
- Classes will mostly be discussion
- We'll rotate leading the discussion
- we'll rotate note taking

Intros and Topics of Interest

- how to understand bias and what can be done, multiple dimension to explore
- more about reading and writing papers
- more skill in reading research papers
- missing data, incomplete problems
- HCI
- breadth, more research
- ML
- eg (pain area)
- noisy data
- natural disaster evacuation plan
- incomplete data
- NLP

Overview of Course Topics

- COMPAS Example
- disparate treatment/impact
- medical
-

Prepare for the next class

Prepare for Wednesday:

Model Based ML: <https://mbmlbook.com/toc.html>

Read: Chapter 1 & the Interlude on the ML life cycle Skim the intro to two application chapters Be prepared to compare this view of ML to how you've learned int (or other CS topics previously)

Read: <https://web.stanford.edu/class/ee384m/Handouts/HowtoReadPaper.pdf>

Be prepared to ask questions about how to prepare for presenting a paper in class

Create or make sure you can log into GitHub Account

2022-01-26

Lead Scribe: Lily

Admin

- sorry about notes
- private github repo → Spring 2022
- grading contract FYI
 - Will be given further instructions on ways to achieve an 'A' or 'B'
 - To get a 'B' you will only need to complete the paper and presentation
 - To get an 'A' you will implement a project (translation)
 - Paper and presentation will be assigned
 - Paper → CS Conference Style
 - Draft due: last day before the presentation, will be posted for the class to review

Opening Question

What kind of data are you most in working with?

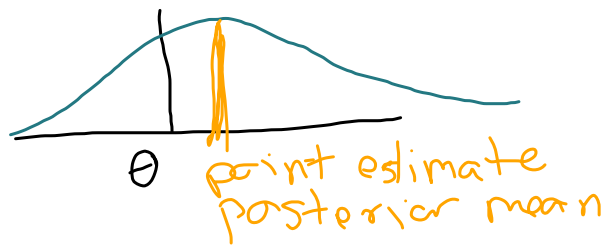
- Class response:
 - GIS data
 - Linguistic data (tweets, reddit posts)
 - Numerical data (tabular)
 - Video/Image
 - Time series
 - EHR/Medical related data
 - NLP
 - tabular/survey

How to Read a Paper

Model Based ML

- Discrete probabilities (distributions introduced in murder mystery chapter)
- Bernoulli
- Priors (probabilistic guess about a random variable)
 - Are useful for working with less data to create strong inferences
 - Working with things when not a lot of data is available
 - Assumptions, expressed in a probability distribution
- Posterior
 - Inference given regularizer: Likelihood...
 - Most common posterior probability distribution we're doing: Probability of parameters given data
- Point Estimate
 - These are the single values produced after training (weights)
 - Posterior mean

$$P(\text{parameters} \mid \text{data})$$



- Most of the probability distributions we'll use belong to the exponential family
 - https://en.wikipedia.org/wiki/Exponential_family
- Conditional Probability
 - One for each value of the conditioning variable

- (e.g.) Murder mystery → murderer variable can be Grey or Auburn

$$P(\text{weapon}) = \sum_{\text{murderer}} P(\text{weapon}, \text{murderer})$$

$$P(w=d) = .03 + .56 = .59$$

$$P(w=r) = .27 + .14 = .41$$

- Marginal probability
 - (Section 1.2 – A Model of Murder)
 - “Probability of one event in the presence of all (or subset) outcomes of the other random variable...” (<https://machinelearningmastery.com/joint-marginal-and-conditional-probability-for-machine-learning/>)
- Maximum Likelihood Estimation
 - Assume a distribution, our goal will be to find the theta (parameter)
 - Maximizing, find parameters that will give us the highest probability (finding the one-parameter—that fits best)
- elicitation - an interdisciplinary field in statistics and psychology; study of how to get an expert's distribution for how likely an event is to occur.

Prepare for next class

- Order of the weekly topics may change
- Dr. Brown will present next week, but we'll start rotating the following week
- There are (2) readings, bring questions and prepare

Learning & Evaluation

- Read through the whole Learning and Evaluation Page after I post a notification to, there are some fixes to be made
- Bring Questions to class next week
- Be ready to work on your grading contract

Reading

The Scientific Method in the Science of Machine Learning and Value-laden Disciplinary Shifts in Machine Learning

Scientific Method & Philosophy of ML

Lead Scribe: Derek Jacobs

admin

- grading contract will be posted in time for Wednesday
- notes & posting [example](#)
- can leave out admin in notes; that material will mostly be other places

Opening Notes

- set the stage for how we think about other work and setting up your projects

Scientific Method in the Science of Machine Learning

[paper](#)

Introduction

- ML is having a hard time explaining some results
- Replications are failing

Scientific Method

Starting from the assumption that there exists accessible ground truth, the scientific method is a systematic framework for experimentation that allows researchers to make objective statements about phenomena and gain knowledge of the fundamental workings of a system under investigation.

—Jessica Zosa Forde and Michela Pagnini

- process and a social contract
 - ML might not be systematic
 - Control the randomness of our algorithms
 - Procedures/Trust allow comparisons of results
- assumes a ground truth
 - Things can go wrong if there simply isn't a ground truth
 - Especially when using ML in social context
- Hypotheses
 - A formed "guess"
 - In CS, the learning process is as follows
 - Here's a problem (that has an answer)
 - Write code to solve it to specifications
 - Hypotheses can be falsified through statistical and other analysis
 - You can prove the opposite is not true, but challenging to prove truth

you can express hypotheses as priors, but that prior wouldn't be the same in the rest of ML lit

Discuss

When might this not apply?

At the base of scientific research lies the notion that an experimental outcome is a random variable, and that appropriate statistical machinery must be employed to estimate the properties of its distribution.

- Treat your accuracies as a random variable as part of an experiment and you're experimenting around those
- In stats people chase having p values ≤ 0.05 and so we have a reproducibility crisis
- [Reproducibility Study](#)
 - < 40% replicated the original results
- there are some problems with NHST but there are alternatives that are still rigorous
 - [for design research](#)
 - [in psych](#)
 - [bayesian in hci](#)
 - Bayesian
 - Priors allowed
 - More broadly defined
 - Everything we do is influenced by our thoughts and so there's some subjectivity
 - Interpreting results in terms of previous results
 - Frequentist

- No Priors
- Probabilities strictly refer to events like fair coin flips (100 flips we expect 50 heads 50 tails)
- Knowledge does not accrue

Case study

HEP:

- proposed theory
- null hypothesis
- careful accounting
- model building and hypothesis testing phases
- parametric models derived from first principles
- statistical test is constructed

ML Analogy:

- suspect new activation
- formulate quantitative hypothesis
 - How will it work, what'll it do, how much will it improve
 - And behavior of how it'll change
- run experiments
 - Record outcomes from base models (no intervention)
- This is a statistical model of your experiment, not an actual model
 - Dataset, optimizer, init, hyperparameters, etc are just noise in the question of "does the activation function work"
 - Or make things specific (improve results with a specific dataset)

Recommendations

1. formulate hypotheses first 2. statistical testing 3. operate in controlled, reproducible, and verifiable settings 4. negative result workshops - [registration workshop at neurips 2021](#) not an author as speaker - If your experimental plan is good, results are published regardless of positive or negative case - [new journal](#)

Value Laden Shifts in ML

[paper](#) @brownsarahm reminder from lily **talk on incorporating ethics into teaching data structs - environmental cost of algorithms**

- Looking at how different values influence what is being done in ML
- disciplinary shifts are not objective, but value laden
- Model Types
 - Different categories of models typically used for specific purposes
 - The structure of what our learning algorithm outputs
 - e.g. CNN for image processing, Linear Models, SVMs, etc

Model types as organizing

- many researchers self-organize in types; eg for reviewing, workshops etc
 - How do we organize them in a package like sklearn, but also "who knows who works on this"
- commitment is fueled by exemplars
- has down stream effects:
 - guide research agenda and problem selection
 - How model types influence problem selection
 - Different types are tailored to different problems
 - Most popular model means most funding
 - More funding on specific models drives improvements to that area specifically and other problems get lost

- constrain search for solutions
- prerequisites, eg deep learning and data volume (fig 1) and compute power (fig2)
 - fig1
 - Depending on data amount, we may favor one model over the other, and that in turn is researched more
 - fig2
 - Shows over time how my computer power is used in training AI
 - Recently exponential growth
- model types have parallels but important differences in philosophical scientific organizing principles
 - decreases theory development
 - Kuhn
 - The organizing paradigm defines what questions are valid
 - For example
 - If we only have earth water fire ether, we can't ask what molecules do
- When committed to model types...
 - The whole industry shifts towards it
 - Like NVidia GPUs, computer architectures, etc

Model type is self-reinforcing

- comparing them is influenced by the model type points above (problems, prerequisites)

Comparison between types is value laden

- Applied to ImageNet
 - Benchmark image classification problem
 - Until 2011, best error rate was 25% (no deep learning)
 - 2012 - AlexNet reached 16% (deep learning)
 - By 2016 ImageNet is basically "done" due to all extremely high accuracies (97% vs 97.1%)
- This doesn't mean the problem of image classification is solved
- prerequisites
 - compute
 - Who has access
 - data
 - Who has access
 - What sets are created/curated
- Evaluation criteria
 - in theories: as a whole, internal consistent, predictive, etc

Evaluating theories based on their theoretical virtues is a value-laden activity when theoretical virtues are carriers of values.

—Milli and Dotan

- eg consistency requires keeping the bad from the old in new
- eg: metrics and discrimination
 - Something that was sexist previously is still now

Conclusion

A related question inspired by these issues is who should make decisions in what values are furthered? Who gets to have a voice? In talking about selection of problems in science, Kitcher (2011) argues that all sides should have a say, including laypersons [71]. A question for machine learning is: is the same true for machine learning? Who should have a say about which criteria are important in evaluating model-types? That is itself another value-laden question.

Overall Paper thoughts

i Discuss

General thoughts?

- Think about things that're interesting, confusing, etc (for future reference)
- Take care when conducting ML
 - Feed in data, hope for the best without considering the "why's"

i Discuss

How is this different from how you've thought about CS before?

- There's more to consider before actually coding
- Vast social influences in things as simple as "can i even get this dataset"
- We focus on what works now instead of what hasn't worked

i Discuss

How might the value-laden points about theory development relate the the scientific method points

- Two pillars of hypothesis + statistical testing

Meta points

- (science) scientific method one is a workshop paper (approx length of your project papers)
- this is a position paper (it's not about new experimental results as much as the classic research arguing a position paper)

next class

- volunteer: Emmely
 - [Roles for computing in social change](<https://dl.acm.org/doi/abs/10.1145/3351095.3372871>)
 - See course site for notes on [expectations during presentations](#)

Roles for Computing in Social Change

Lead Scribe: Damon Coffey

Roles for Computing in Social Change -concerns about fairness, bias and accountability in the field

Introduction:

- high stakes decision making algorithms have potential to predict outcomes more accurately
- cs has generally failed to target the correct point of intervention

- ex: intervention at the selection phase in an employment context could prevent a hostile work place

Computing as a Diagnostic

- computing can help us measure social problems and diagnose how they manifest in tech systems
- computing cannot solve issues on its own
- Diagnostics work can be valuable
 - highlight tech dimensions of social problems
- misinformation can negatively affect marginalized populations more ex: search engines displaying low quality health information
- not presented as solutions, rather as tools to document practices
 - not to confuse diagnostics with treatment
 - computing is not unique in helping diagnose social problems
 - sociology, etc..
 - certain tools can be treated as certainty for every situation, which is not the case

Computing as a Formalizer

- computing requires explicit specification of inputs and goals
- these inputs and goals can be affected by transparency, accountability and stake holder participation
 - need to be precise ex: risk assessment: debate over how to formalize pretrial risk, if and how to use these instruments
- not all data is easy to quantify
- may press people to rely on measures that are incorrect

Computing as Rebuttal

- computing can clarify the limits of technical interventions and of policies promised on them
- limits of computing can drive people to reject computational approaches
- ex: using an algorithm to determine an immigrant's societal worth, not good. Should seek a different method rather than forcing a technological one
- need to understand what algorithms are actually capable of, instead of forcing it on everything
 - need to show what an algorithm CANT do (prove limits)
- prediction algorithms for risk assessment
- computational research on fairness is built on discrimination law
- Risks
 - proclamations of what a computational tool is incapable of may focus on improving tool even if it is not possible

Computing as a Synecdoche

- computing can foreground long standing social problems in a new way
- Eubank's core concern: computing is just one mechanism through which longstanding poverty policy is manifested
- Automated systems can divert poor people from the resources they need
- computing can help bring attention to old problems, however
- synecdochal focus on computing must walk a pragmatic line between over emphasis on tech aspects and recognition of the work tech actually does
- need to find a balance between the two and develop better systems with more emphasis on social issues

Missing data Intro

Lead Scribe: Surbhi Rathore

Opening Notes

Paper - Handling Missing Values when Applying Classification Models

[paper](#) – Presented by Emmely

Analysis over different treatments of missing values at prediction time.

- Alternative courses of action when features are missing:
 - Discard instances: Simply discarding instances with missing values.
 - Acquire missing values: In practice, a missing value may be obtainable by incurring a cost, such as the cost of performing a diagnostic test or the cost of acquiring consumer data from a third party.
 - Imputation : The main idea of imputation is that if an important feature is missing for a particular instance, it can be estimated from the data that are present.
 - (Predictive) Value Imputation (PVI): Value imputation estimates a value to be used by the model in place of the missing feature. Value imputation is more common in the statistics community
 - Distribution-based Imputation (DBI): Distribution-based imputation estimates the conditional distribution of the missing value, and predictions will be based on this estimated distribution. distribution-based imputation is the basis for the most popular treatment used by the (non-Bayesian) machine learning community
 - Unique-value imputation: Rather than estimating an unknown feature value it is possible to replace each missing value with an arbitrary unique value.
 - Reduced-feature Models: Reduced-feature models : We refer to these models as reduced-feature models, as they are induced using only a subset of the features that are available for the training data. Reduced-feature models can be computationally expensive.
- Missing Completely At Random (MCAR) : refers to the scenario where missingness of feature values is independent of the feature values (observed or not). (missing values have nothing to do with feature values)

Comparison of PVI, DBI and Reduced Modeling

- Reduced-feature modeling performs consistently outperforms the other two method.

LOW and HIGH FEATURE IMPUTABILITY Result

- PVI is better for higher feature imputability, and DBI is better for lower feature imputability. Value imputation generally preferable for high feature imputability, and DBI generally better for low feature imputability.

REDUCED-FEATURE MODELING SHOULD HAVE ADVANTAGES ALL ALONG THE IMPUTABILITY SPECTRUM

- Reduced modeling is a lower-dimensional learning problem than the modeling to which imputation methods are applied, it will tend to have lower variance and thereby may exhibit lower generalization error.

Evaluation with “Naturally Occurring” Missing Values

- By “naturally occurring,” we mean that these are data sets from real classification problems, where the missingness is due to processes of the domain outside the control.

Conclusions

Reduced-feature models are preferable both to distribution-based imputation and to predictive value imputation. Reduced models undertake a lower-variance learning task, and do not fall prey to certain pathologies. Predictive value imputation and DBI are easy to apply, but one almost always pays—sometimes dearly—with suboptimal accuracy.

Paper - Missing data imputation using statistical and machine learning methods in a real breast cancer problem

[paper](#) – Presented by Chan

The “El Alamo-I” breast cancer dataset

- one of the largest databases on breast cancer in Spain.
- The missing data represent 5.61% of the overall data set.

Statistical methods

- The statistical imputation methods include mean imputation, hot-deck, and MI methods based on regression and the expectation maximisation (EM) algorithm.
 - Mean Imputation: is a simple application of regression imputation, the mean value of each non-missing variable is used to fill in missing values for all observations.
 - Hot-deck imputation : nearest neighbour hot-deck imputation is applied, where a nonrespondent is assigned the value of the nearest neighbour record according to a similarity criterion.
 - Multiple imputation : , MI replaces an unknown value with a set of plausible data and uses an appropriate model that incorporates random variation. MI has several desirable features:
 - (1) an appropriate random error is introduced into the imputation process to obtain approximately unbiased estimates of all parameters;
 - (2) good estimates of standard errors are obtained from repeated imputation; and
 - (3) MI can be used for any kind of data and any kind of analysis without specialised software.

Machine learning methods

- Imputation methods based on machine learning are sophisticated procedures that generally consist of creating a predictive model to estimate values that will substitute the missing items.
- These approaches model the missing data estimation based on information available in the data set.
- Three well-known imputation techniques using machine learning approaches: MLP, SOM and KNN.
 - Multi-layer perceptron :
 - An MLP consists of multiple layers of computational units interconnected in a feed-forward way.
 - Each unit in one layer is directly connected to the neurons of the subsequent layer.
 - A fully connected, two-layered MLP architecture was used and sigmoidal activation functions.
 - MLP networks can be used to estimate missing values by training an MLP to learn the incomplete features (used as outputs), using the remaining complete features as inputs.
 - Self-organisation maps :
 - An SOM is a neural network model made out of a set of nodes (or neurons) that are organised on a 2D grid and fully connected to the input layer.
 - The training of a basic SOM is performed using an iterative process. After the weight vectors are initialised, they are updated using all input training vectors.
 - After the SOM model has been trained, it can be used to estimate missing values.
 - K-nearest neighbours :
 - the KNN imputation algorithm uses only similar cases with the incomplete pattern.
 - Given an incomplete pattern x , this method selects the K closest cases that are not missing values in the attributes to be imputed (i.e., features with missing values in x), such that they minimise some distance measure.
 - The optimal value of K is usually chosen by crossvalidation.
 - Calculation of replacement value depends on the type of data;
 - example, the mode is selected for discrete data (categorical data),
 - while the mean is used for numerical data (continuous data).
- The ANN prognosis model :

- numerical simulations are performed on the data imputed by the methods described above on neural networks comprising a single hidden layer with the number of neurons between 2 and 50.

Model evaluation

- The accuracy of the prognosis models is evaluated by testing two main properties:
 - discrimination and calibration. Discrimination is the ability to separate patients with and without a relapse event.
 - Calibration is the ability to correctly estimate the risk or probability of a future event.

Conclusions

machine learning techniques may be the best approach to imputing missing values, as they led to statistically significant improvements in prediction accuracy. Imputation techniques depend on the available data and the prediction model used. Also, these results might not generalise to different data sets

closing

- Volunteers for paper presentation: Chamudi & Lily

MLSS

2022-02-09

Lead Scribe: Chamudi Kashmila

Paper - Graphical Models for Inference with Missing Data

[paper](#) – Presented by Chamudi

Graphical Models for Inference with Missing Data

Missing data can have several harmful consequences.

- Firstly they can significantly bias the outcome of research studies.
 - This is mainly because the response profiles of non-respondents and respondents can be significantly different from each other. Hence ignoring the former distorts the true proportion in the population.
- Secondly, performing the analysis using only complete cases and ignoring the cases with missing values can reduce the sample size there by substantially reducing estimation efficiency.
- Lastly, many of the algorithms and statistical techniques are generally tailored to draw inferences from complete datasets.
 - it may be difficult or even inappropriate to apply these algorithms and statistical techniques on incomplete datasets.

Existing Methods for Handling Missing Data

- Listwise deletion (LD) and pairwise deletion (PD) are used in approximately 96% of studies in the social and behavioral sciences
- Expectation-maximization (EM) [example: K-Means] algorithm is a general technique for finding max likelihood (ML) estimates from incomplete data.
- ML is often used in conjunction with imputation methods
 - Mean Sub, Hot-deck imputation, cold-deck imputation, and Multiple Imputation (MI).
- This paper aims to illuminate missing data problems using causal graphs
- The objectives are:

- Given a target relation Q to be estimated and a set of assumptions about the missingness process encoded in a graphical model, (i) under what conditions does a consistent estimate exist, and (ii) how can we produce it from the data available. These questions are answered with the aid of **Missingness Graphs (m-graphs)**
- Review the traditional taxonomy of missing data problems and cast it in graphical terms.
- define the **notion of recoverability** - the existence of a consistent estimate - and present graphical conditions for detecting recoverability of a given probabilistic query Q .

Graphical Representation of the Missingness Process

- Graphical Models have been used to analyze missing missing information in the form of missing cases
- The need exists for a general approach capable of modeling an arbitrary data-generating process and deciding whether how missingness can be outmaneuvered in every dataset generated by that process.
- Such a general approach should allow each variable to be governed by its own missingness mechanism, and each mechanism to be triggered by other partially observed variables in the model.
- To achieve this flexibility we use a graphical model called **missingness graph** which is a **DAG (Directed Acyclic Graph)** defined as follows.

Missingness Graphs

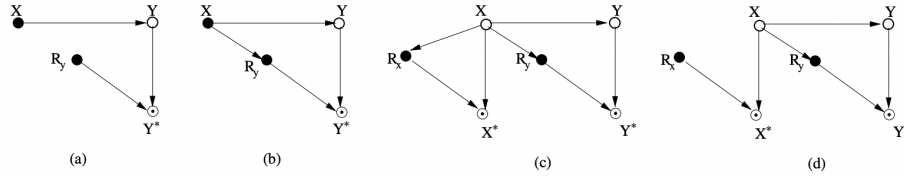


Figure 1: m -graphs for data that are: (a) MCAR, (b) MAR, (c) & (d) MNAR; Hollow and solid circles denote partially and fully observed variables respectively.

Let $G(\mathbb{V}, E)$ be the causal DAG where $\mathbb{V} = V \cup U \cup V^* \cup R$

V is the set of observable nodes. Nodes in the graph correspond to variables in the data set. U is the set of unobserved nodes E is the set of edges in the DAG V^* is a set of all proxy variables R is the set of all causal mechanisms that are responsible for missingness

- Oftentimes we use bi-directed edges as a shorthand notation to denote the existence of a U variable as common parent of two variables in $V_o \cup V_m \cup R$.
- V is partitioned into V_o and V_m such that
- $V_o \subseteq V$ is the set of variables that are observed in all records in the population
- $V_m \subseteq V$ is the set of variables that are missing in at least one record.
- Variable X is termed as **fully observed** if $X \in V_o$ and **partially observed** if $X \in V_m$.

Associated with every partially observed variable $V_i \in V_m$ are two other variables R_{v_i} and V_i^* , where V_i^* is a proxy variable that is actually observed, and R_{v_i} represents the status of the causal mechanism responsible for the missingness of V_i^* ; formally,

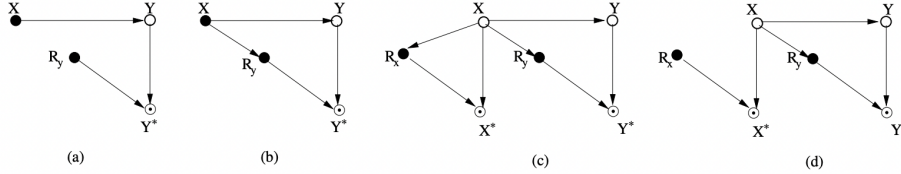
$$v_i^* = f(r_{v_i}, v_i) = \begin{cases} v_i & \text{if } r_{v_i} = 0 \\ m & \text{if } r_{v_i} = 1 \end{cases} \quad (1)$$

- This graphical representation briefly shows both the causal relationships among variables in V and the process that accounts for missingness in some of the variables.
- Since every d-separation in the graph implies conditional independence in the distribution, the m -graph provides an effective way of representing the statistical properties of the missingness process and, hence, the potential of recovering the statistics of variables in V_m from partially missing data.

Taxonomy of Missingness Mechanisms

It is common to classify missing data mechanisms into three types

- **Missing Completely At Random (MCAR)** : Data are MCAR if the probability that V_m is missing is independent of V_m or any other variable in the study, as would be the case when **respondents decide to reveal their income levels based on coin-flips**.
- **Missing At Random (MAR)** : Data are MAR if for all data cases Y , $P(R|Y_{obs}, Y_{mis}) = P(R|Y_{obs})$ where Y_{obs} denotes the observed component of Y and Y_{mis} , the missing component. Example: **Women in the population are more likely to not reveal their age**.
- **Missing Not At Random (MNAR)** or “non-ignorable missing”: Data that are neither MAR nor MCAR are termed as MNAR. Example: **Online shoppers rate an item with a high probability either if they love the item or if they dislike it. In other words, the probability that a shopper supplies a rating is dependent on the shopper’s underlying liking**.



- In the graph-based interpretation used in this paper, MCAR is defined as total independence between \mathbf{R} and $\mathbf{V}_o \cup \mathbf{V}_m \cup \mathbf{U}$ i.e. $\mathbf{R} \perp\!\!\!\perp (\mathbf{V}_o \cup \mathbf{V}_m \cup \mathbf{U})$, as shown in Figure a.
- MAR is defined as independence between \mathbf{R} and $\mathbf{V}_m \cup \mathbf{U}$ given \mathbf{V}_o i.e. $\mathbf{R} \perp\!\!\!\perp \mathbf{V}_m \cup \mathbf{U} | \mathbf{V}_o$, as shown in Figure b.
- Finally if neither of these conditions hold, data are termed MNAR, as shown in Figure c and d.

Recoverability

Recoverability is a measurement to see if we can get $P(X, Y)$ from the entire dataset D . Examine the conditions under which a bias-free estimate of a given probabilistic relation Q can be computed.

Definition (Recoverability). Given a **m-graph** G , and a **target relation** Q defined on the variables in \mathbf{V} , Q is said to be **recoverable** in G if there **exists an algorithm that produces a consistent estimate of Q for every dataset D such that $P(D)$ is (1) compatible with G and (2) strictly positive over complete cases** i.e. $P(\mathbf{V}_o, \mathbf{R} = 0) > 0$.

Recoverability when data are MCAR

For MCAR data we have $\mathbf{R} \perp\!\!\!\perp (\mathbf{V}_o \cup \mathbf{V}_m)$. Therefore, we can write $P(V) = P(V | \mathbf{R}) = P(\mathbf{V}_o, \mathbf{V}^* | \mathbf{R} = 0)$.

Since both \mathbf{R} and \mathbf{V}^* are observables, the joint probability $P(\mathbf{V})$ is **consistently estimable (recoverable)** by considering complete cases only

Example : Let X be the treatment and Y be the outcome as depicted in the m-graph in Fig. 1 (a). Let it be the case that we accidentally deleted the values of Y for a handful of samples, hence $Y \in \mathbf{V}_m$. Can we recover $P(X, Y)$?

From D , we can compute $P(X, Y, R_y)$. From the m-graph G , we know that Y is a collider and hence by d-separation, $(X \cup Y) \perp\!\!\!\perp R_y$. Thus $P(X, Y) = P(X, Y | R_y)$. In particular, $P(X, Y) = P(X, Y | R_y = 0)$. When $R_y = 0$, by eq. (1), $Y = Y^*$. Hence, $P(X, Y) = P(X, Y^* | R_y = 0)$ (2) The RHS of Eq. 2 is consistently estimable from D ; hence $P(X, Y)$ is recoverable.

Recoverability when data are MAR

When data are **MAR**, we have $\mathbf{R} \perp\!\!\!\perp \mathbf{V}_m | \mathbf{V}_o$. Therefore $P(V) = P(\mathbf{V}_m | \mathbf{V}_o)P(\mathbf{V}_o) = P(\mathbf{V}_m | \mathbf{V}_o, \mathbf{R} = 0)P(\mathbf{V}_o)$. Hence the joint distribution $P(\mathbf{V})$ is recoverable.

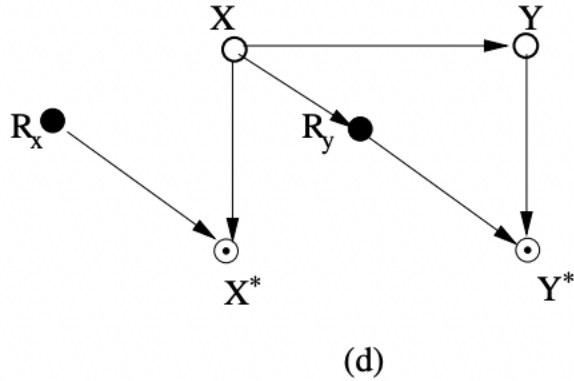
Example : Let X be the treatment and Y be the outcome as shown in the m-graph in Fig. 1 (b). Let it be the case that some patients who underwent treatment are not likely to report the outcome, hence the arrow $X \rightarrow R_y$. Under the circumstances, can we recover $P(X, Y)$?

From D , we can compute $P(X, Y, R_y)$. From the m-graph G , we see that Y is a collider and X is a fork. Hence by d-separation, $Y \perp\!\!\!\perp R_y | X$. Thus $P(X, Y) = P(Y | X)P(X) = P(Y | X, R_y)P(X)$. In particular, $P(X, Y) = P(Y | X, R_y = 0)P(X)$. When $R_y = 0$, by eq. (1), $Y^* = Y$. Hence, $P(X, Y) = P(Y^* | X, R_y = 0)P(X)$ (3) and since X is fully observable, $P(X, Y)$ is recoverable

Recoverability when data are MNAR

Data that are neither MAR nor MCAR are termed **MNAR**. Though it is generally believed that relations in MNAR datasets are not recoverable, the following example demonstrates otherwise.

Example : Fig. 1 (d) depicts a study where (i) some units who underwent treatment ($X = 1$) did not report the outcome (Y) and (ii) we accidentally deleted the values of treatment for a handful of cases. Thus we have missing values for both X and Y which renders the dataset MNAR. We shall show that $P(X, Y)$ is recoverable.



From D, we can compute $P(X^*, Y^*, R_x, R_y)$. From the m-graph G, we see that $X \perp\!\!\!\perp R_x$ and $Y \perp\!\!\!\perp (R_x \cup R_y) | X$.

Thus $P(X, Y) = P(Y | X)P(X) = P(Y | X, R_y = 0, R_x = 0)P(X | R_x = 0)$. When $R_y = 0$ and $R_x = 0$ we have (by Equation (1)), $Y = Y^*$ and $X = X^*$.

Hence, $P(X, Y) = P(Y^* | X^*, R_x = 0, R_y = 0)P(X^* | R_x = 0)$ (4) Therefore, $P(X, Y)$ is recoverable

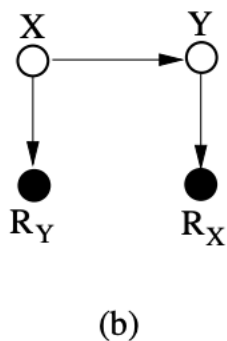
Conditions for Recoverability

How can we determine if a given relation is recoverable? **Theorem 1:** A query Q defined over variables in V_0 Theorem 1 provides a sufficient condition for recoverability

Theorem 1 A query Q defined over variables in $V_0 \cup V_m$ is recoverable if it is decomposable into terms of the form $Q_j = P(S_j | T_j)$ such that T_j contains the missingness mechanism $R_v = 0$ of every partially observed variable V that appears in

Proof: If such a decomposition exists, every Q_j is estimable from the data, hence the entire expression for Q is recoverable.

Example : Consider the problem of recovering $Q = P(X, Y)$ from the m-graph of Fig. 3(b).



Attempts to decompose Q by the chain rule, as was done in Eqs. (3) $P(X, Y) = P(Y^* | X, R_y = 0)P(X)$ (3) and (4) would not satisfy the conditions of Theorem 1.

To witness we write $P(X, Y) = P(Y | X)P(X)$ and note that the graph does not permit us to augment any of the two terms with the necessary R_x or R_y terms;

X is independent of R_x only if we condition on Y, which is partially observed, and Y is independent of R_y only if we condition on X which is also partially observed.

This deadlock can be disentangled however using a non-conventional decomposition:

$$\begin{aligned} Q = P(X, Y) &= P(X, Y) \frac{P(R_x, R_y | X, Y)}{P(R_x, R_y | X, Y)} \\ &= \frac{P(R_x, R_y) P(X, Y | R_x, R_y)}{P(R_x | Y, R_y) P(R_y | X, R_x)} \end{aligned} \quad (5)$$

where the denominator was obtained using the independencies $R_x \perp\!\!\!\perp (X, R_y) | Y$ and $R_y \perp\!\!\!\perp (Y, R_x) | X$ shown in the graph.

The final expression above satisfies **Theorem 1** and renders **P(X,Y) recoverable**.

This example again shows that **recovery is feasible** even when data are **MNAR**.

Theorem 2 operationalizes the decomposability requirement of *Theorem 1*.

Theorem 2

(Recoverability of the Joint P(V)). Given a m-graph G with no edges between the R variables and no latent variables as parents of R variables, a necessary and sufficient condition for recovering the joint distribution P(V) is that no variable X be a parent of its missingness mechanism R_x . Moreover, when recoverable, P(V) is given by

$$P(v) = \frac{P(R = 0, v)}{\prod_i P(R_i = 0 | pa_{r_i}^o, pa_{r_i}^m, R_{pa_{r_i}^m} = 0)}, \quad (6)$$

where $Pa_{r_i}^o \subseteq V_o$ and $Pa_{r_i}^m \subseteq V_m$ are the parents of R_i .

Theorem 3 gives a sufficient condition for recovering the joint distribution in a Markovian model.

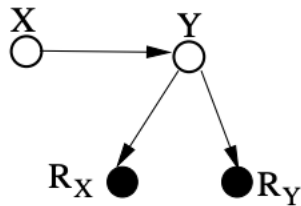
Theorem 3 Given a m-graph with no latent variables (i.e., Markovian) the joint distribution P(V) is recoverable if no missingness mechanism R_x is a descendant of its corresponding variable X. Moreover, if recoverable, then P(V) is given by

$$P(v) = \prod_{i, V_i \in V_o} P(v_i | pa_i^o, pa_i^m, R_{pa_i^m} = 0) \prod_{j, V_j \in V_m} P(v_j | pa_j^o, pa_j^m, R_{V_j} = 0, R_{pa_j^m} = 0), \quad (9)$$

where $Pa_i^o \subseteq V_o$ and $Pa_i^m \subseteq V_m$ are the parents of V_i .

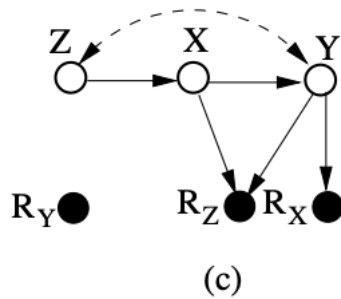
Theorem 4 A sufficient condition for recoverability of a relation Q is that Q be decomposable into an ordered factorization, or a sum of such factorizations, such that every factor $Q_i = P(Y_i | X_i)$ satisfies $Y_i \perp\!\!\!\perp (R_{Y_i}, R_{X_i}) | X_i$. A factorization that satisfies this condition will be called admissible.

Theorem 4 will allow us to confirm recoverability of certain queries Q in models such as those in Figures a, c, d which do not satisfy the requirement in *Theorem 2*

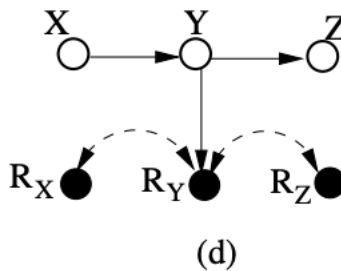


(a)

$P(X|Y) = P(X|R_X = 0, R_Y = 0, Y)$ is recoverable



$P(X, Y, Z) = P(Z|X, Y, R_Z = 0, R_X = 0, R_Y = 0) P(X|Y, R_X = 0, R_Y = 0) P(Y|R_Y = 0)$ is recoverable



$P(X, Z) = P(X, Z|R_X = 0, R_Z = 0)$ is recoverable

Conclusion

- Causal graphical models depicting the data generating process can serve as a powerful tool for analyzing missing data problems
- Formalized the notion of recoverability and showed that relations are always recoverable when data are missing at random (MCAR or MAR) and even when data are missing not at random (MNAR).
- presented a sufficient condition to ensure recoverability of a given relation Q (Theorem 1) and operationalized Theorem 1 using graphical criteria (Theorems 2, 3 and 4).

Fairness

Lead Scribe: Chan

FAIRNESS AND MACHINE LEARNING

Chapter 2: Classification

Presented by Damon

Main Goal

- determine plausible value for an unknown variable Y given an observed variable X
- Essentially trying to predict something based on other factors

Supervised learning

- The prevalent method for constructing classifiers from observed data
 - Goal: to identify meaningful patterns
- Classifier: a mapping from the space of possible values for X to the space of values that the target Y can assume.
- The essential idea is simple in that we will have labeled data in the form (x_1, y_1) that is drawn from a distribution where x_1 is an instance and y_1 is the label.
 - The instances are then partitioned into positive

- Labels typically come from a discrete set such as $\{-1, 1\}$ in the case of binary classification.

Statistical Classification Criteria

- How to identify which classifier is best for your purpose
- Accuracy: the probability of correctly predicting the target variable
- Cond. probability table
 - The true positive rate corresponds to the frequency of the classifier correctly assigning a positive label to a positive instance.
- Why do we need the weighted average?
 - $P(Y=\hat{Y}) = P(\hat{Y}=1|Y=1)P(Y=1) + P(\hat{Y}=0|Y=0)P(Y=0)$
 - to get all of the accuracy, we need to multiply the weights to our conditional probabilities so that it is computable.

The Conditional Expectation

- **Score:** a single real-valued variable summarized from a regression model
- A natural score function is the expectation of the target variable Y conditional on the features X we have observed

Sensitive Characteristics (A)

- In many classification tasks, the features X contain or implicitly encode sensitive characteristics of an individual
- It is dangerous to ignore these factors, which will make prediction difficult by tampering the correlation.

Independence – $R \perp A$

- Requires the sensitive characteristic to be statistically independent of the score
- **Definition 1:** The random variable (A, R) satisfy independence if $A \perp R$
- Independence simplifies to the condition:
 - $P\{R=1|A=a\} = P\{R=1|A=b\}$
- Where $R=1$ is acceptance and the condition requires the acceptance rate to be the same in all groups.
- There could also be a relaxation on the constraint that would introduce a positive amount of slack ϵ and require that:
 - $P\{R=1|A=a\} \geq P\{R=1|A=b\} - \epsilon$
 - $\epsilon = P\{R=1|A=b\} - P\{R=1|A=a\}$, where we want the difference to be small.

Separation

- Acknowledges that in many scenarios, the sensitive characteristic may be correlated with the target variable.
- This separation criterion allows correlation between the score and the sensitive attribute to the extent that it is justified by the target variable
- **Definition 2:** Random variables (R, A, Y) satisfy separation if $R \perp A | Y$
- In the case where R is a binary classifier, separation is equivalent to requiring for all groups a, b the two constraints:
 - $P\{R=1|Y=1, A=a\} = P\{R=1|Y=1, A=b\}$
 - $P\{R=1|Y=0, A=a\} = P\{R=1|Y=0, A=b\}$
- Separation requires that all groups experience the same false negative rate and the same false positive rate.

$$\bullet \quad \frac{TP_a}{TP_a + FN_a} = \frac{TP_b}{TP_b + FN_b}$$

$$\bullet \quad \frac{FP_a}{FP_a + TN_a} = \frac{FP_b}{FP_b + TN_b}$$

Sufficiency – $Y \perp A | R$

- Formalizes that the score already includes the sensitive characteristic for the purpose of predicting the target
- **Definition 3:** We say that random variables (R, A, Y) satisfy sufficiency if $Y \perp A | R$
- In this case, a random variable R is sufficient for A iff for all groups a, b and all values r in the support of R , we have

$$\mathbb{P}\{Y = 1 \mid R = r, A = a\} = \mathbb{P}\{Y = 1 \mid R = r, A = b\}$$

- When R only has 2 values we recognize this condition as requiring a parity of pos./neg. predictive values across all groups.

$$\frac{TP_a}{TP_a + FP_a} = \frac{TP_b}{TP_b + FP_b}$$

$$\frac{FN_a}{FN_a + TN_a} = \frac{FN_b}{FN_b + TN_b}$$

Calibration and Sufficiency

- It is sometimes desirable to be able to interpret the values of the score functions as probabilities
- This condition means that the set of all instances assigned a score value r has an r fraction of positive instances among them.

Calibration by Group

- To formalize the connection between sufficiency and calibration we say that the score R satisfies calibration by group if it satisfies:

$$\mathbb{P}\{Y = 1 \mid R = r, A = a\} = r$$

- **Proposition 1.** If a score R satisfies sufficiency, then there exists a function $l: [0,1] \rightarrow [0,1]$ so that $l(R)$ satisfies calibration by group.

Relationships Between Criteria

- Initial criteria constrains the joint distribution in non-trivial ways therefore we can suspect that imposing any 2 of them simultaneously over-constrains the space to the point where only degenerate solutions remain.
- Happy path of establishing multiple fairnesses is intangible.

Machine Bias

Presented by Derek

The premise

- Brisha Borden & a friend - 18 years old
 - Scooter
- Vernon Prater - "Seasoned"
 - Shoplifting
- Both similar crimes
- Computer program predicting likelihood of recidivism
 - Borden > Prater
 - Two years later, opposite held true

Risk Assessments

- Becoming increasingly more popular in courtrooms
- 9 states provide scores to judges during sentencing

Following the Warnings of AG Holder

- The sentencing commission did not launch a study of risk scores
- ProPublica did launch a study

- >7000 people arrested in Broward Cty, FL
- Check how many were charged with new crimes over 2 years
- Results
 - Unreliable
 - Biased

In Theory

- Risk scores are great
- High scores mean more likely to commit crime
- Vice versa for low scores
- Simple...?

In Practice

- People are extremely complex
- Countless factors go into recidivism such as
 - employment
 - housing status
 - financial situations
- As a result unless we individualize the risk assessment, result will be biased

The Problem

- people tend to blindly trust tech.
- any jurisdictions adopt Northpointe's software without testing
- Using tech is the easy way out
 - "Easy to use and gives 'simple/effective' charts for judicial review"
- 2009 study reported a 68% accuracy rate for the scoring software

Overall

- Using tech in social context is extremely complex
- People are unique and countless factors go into individual behavior
- With recidivism specifically
 - Black people more likely to be predicted higher than White people
 - Software relied upon too much
 - Studies have concluded risk scores do not reflect reality.

Gender Shades

Presented by Derek

Intro

- AI is widely used in society
- Facial recognition software can realistically be used to identify suspects
- Algo. trained with biased data result in algorithmic discrimination
- This work focuses on facial recognition to gender classification

Related Work

- Automated Facial Analysis
- Benchmarks
- Quality of models

Intersectional Benchmark

- Gender classification means we need defined classes
- The dataset should contain varied physical attributes for subgroup accuracy analysis
- Pilot Parliaments Benchmarks dataset
 - high quality photos
 - reliable sources

Commercial Gender Classification Audit

- Classifiers perform best on lighter, male faces
- Classifiers perform worst on darker female faces
 - Microsoft, IBM, Face++ analyzed
- Max disparity in error rate between best/worst classified groups is 34.4%

Dissecting racial bias in an algorithm used to manage the health of populations

Presented by Derek

- Black people had to have more chronic conditions to have a higher risk score
- Training used total medical expenditure.
 - Could have used total sickness as a factor to train to avoid bias.

2022-02-23

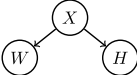
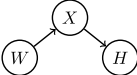
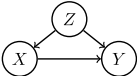
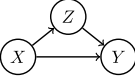
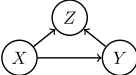
Lead Scribe : Derek

Fairness and Causality/Experiments

Causality

Presented by Surbhi

- Main Goal
 - Teach more how causality can help us understand a situation
- Passive observation
 - Something we generally observe but it does not require proper attention
 - Like noticing a car driving by you
 - Data collected this way shows a snapshot of the world as it is
- Causal reasoning
 - Important questions are not usually observational in nature
 - Would traffic fatalities decrease if we raised the legal driving age by 2 years?
- Not every cause question is easy to address
 - Causal inference gives a formal language to ask questions, but does not trivialize finding the answer
- Supporting 3 purposes for understanding causality
 1. Conceptualize and address limits of observational techniques
 2. Provide tools to help design intervention that achieve a desired effect
 3. Engage with important debate about when and how much reasoning about discrimination and fairness require causal understanding
- Simpson's Paradox
 - Higher acceptance ratio among women, while two show a higher acceptance for men
 - There's variation between departments, and so the higher rate for men may be seen in aggregate but reversed at the department level
 - This is one example of us misinterpreting what conditional probabilities encode
 - If we look at overall data, it looks like admissions are unfair and discriminating for men, but broken by department it looks like it's the other way around
 - Not the admission that makes it look this way but it depends on where women/men tend to apply and such
 - For example engineering vs english vs other
- Causal Models
 - Causal inference can be used to guide design of studies
 - Choosing which variables to include, which to exclude, which to hold constant
 - Serve as mechanism to incorporate scientific domain knowledge and exchange plausible conclusions
- Structural Causal Models
 - A sequence of assignments for generating a joint distribution from independent noise models
 - By executing the sequence of assignments, we build a set of jointly distributed random variables

- Provides a joint distribution and describes how the dist can be generated from elementary noise variables
 - A Probabilistic model
 - Different set of operations
 - Useful for considering hypothetical scenarios differently than if you have a single set of data
 - How we can intervene in a world AND measure the effect of that intervention
 - Causal Graphs
 - Parent notation we saw last week, interpreted a little differently and we can take a model to build a graph
 - Can go from graph to model too
 - Looking at directed graphs as placeholder for an unspecified structural causal model that has the assignment structure given by the graph
- 

- Graph Structure
 - Forks
 - Has outgoing edges to two other variables
 - Aka the most common cause of other variables
- 
- Mediators
 - A case of a fork where x causes z and z causes y so x causes y multiple ways
- 
- Colliders
 - Not confounders
 - X and Y are unconfounded, and we can replace do-statements by conditional probabilities
- 
- The Harvard Admission example
 - The story
 - Asian American Male 25% admission chance, as white 36%, as Hispanic 77%, as African American 95%
 - Invalid statistically because everything on the application is exactly the same except for race but typically many other things will change besides just the race

Empirical Comparison

Presented by Chan

- Main Goal
 - Metaanalysis paper of analyzing fair machine learning algorithms
 - Comparison of algorithms they'd found
- Problem
 - Fairness
 - Comes with sensitive data
 - Age, race, gender, etc
- The Experiments
 - Testing on
 - Preprocessing
 - Comes in 2 ways before feeding into the algorithms that may or may not preprocess further
 - Training data is the cause of discrimination motivates this
 - If training data is discriminating then the results definitely will be, so we can preprocess to make it more fair
 - Modifications to the algorithms
 - Postprocessing

- If we have a result, can we prune or round items for better results
- The Results
 - Measures of discrimination correlate with each other
 - Algorithms make different fairness accuracy tradeoffs
 - Algorithms are fragile: they are sensitive to variations in the input
- Datasets
 - Ricci
 - Determining if firefighters would receive a promotion
 - Adult Income
 - Predicting income above or below \$50k
 - German
 - Classifying people on good or bad credit risk
 - ProPublica Recidivism/Violent Recidivism
 - Committing a crime/violent again
- Preprocessing
 - Modifying input according to any data-specific needs
 - Removing unneeded features, imputing missing data, etc
 - Add a combined sensitive attribute i.e. "White-Woman"
 - Sensitive Attribute treated as binary
 - Analysis
 - See [Figure 2](#) in the paper
 - One dot is one fold (LOOCV) of the data score in whatever metric in one algorithm
 - Equality between preprocessing choices = Showing the plot is not square, where the x and y axis are equal
 - Binary sensitive attribute is more accurate when using a fair classifier
 - Things we knew about ML before don't necessarily apply to fair classifiers
 - Fairness may limit our accuracy

2022-02-28

Lead Scribe: Damon

Elastic Net

Presented by Alex

Introduction

- Ordinary Least Squares (OLS) often poor at prediction and interpretation
- need penalization techniques to help
 - Ridge Regression
 - Best Subset Selection
 - Lasso
 - None of these 3 techniques dominate the other 2, however

Lasso

- a penalized least squares method imposing an L1-penalty on the regression coefficients
- Advantage: produces a sparse representation, so more appealing than ridge and best subset
 - better at getting rid of noise
- Limitations:
 - in $p > n$ case, lasso selects at most n variables before it saturates
 - also, the lasso is not well defined unless the bound on the L1-norm of the coefficients is smaller than a certain value
 - also, if there are high correlations, ridge is better

Paper Goals

- find a new method that works as well as lasso whenever lasso is best, but fixes issues highlighted above when it needs to

Introduces elastic net, like "improved Lasso"

- simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables
- Hybrid Elastic-Net Regression is good at dealing with situations when there are correlations between parameters
- Elastic net is basically a combination of ridge and lasso, but better. Often outperforms lasso
- The elastic net produces a sparse model with good prediction accuracy, while encouraging a grouping effect.

Elastic Net Penalization

- regular regression cost function with L2 and L1 added with coefficients
- the elastic net penalty can be used in classification problems with any consistent loss functions, including the L2-loss which we have considered here and binomial deviance.

Naive Elastic Net

- Impact of alpha
 - if $\alpha = 1$ -> ridge regression
 - $\alpha = 0$ -> lasso regression
 - $0 < \alpha < 1$ -> elastic net
- Why Naive
 - in the procedure for finding elastic nets method, 2 stages involving both lasso and regression techniques
 - first find the ridge regression coefficients, and then do the lasso-type shrinkage along the lasso coefficient solution path

Examples

- Predicting likelihood of prostate cancer example
 - Out of the the 5 methods tested, elastic net is best, OLS is worst
 - Also, naive elastic net performs identically to the ridge regression, fails to do variable selection

Results/Takeaways

- In all examples, elastic net is significantly more accurate than lasso, even when lasso is doing much better than ridge regression
- Elastic net also produce sparse solutions, elastic tends to select more variables than lasso due to grouping effect
- In general, elastic net > lasso
- Summary
 - elastic net method performs variable selection and regularization simultaneously
 - they view the elastic net as a generalization of the lasso, which has been shown to be a valuable tool for model fitting and feature extraction

2022-03-02

Lead Scibe: Derek

Multi Objective in Data Mining

Presented by Chamudi

Intro

- Model Quality represented by an n-dimensional vector
 - n is the num of qual criteria to be optimized
- 2 scenarios
 - Predictive tasks
 - Classificaiton, regression, etc

- Which model is better
 - Easily interpretable dozen node decision tree with 92% test accuracy or non-interpretable hundred node decision tree with 95%
 - The answer depends on the problem at hand and what the user prefers
 - May be after just accuracy or you may want to better understand what's going on
 - One may overfit/underfit, etc
 - Alex's MNIST example → for another class accuracy was the end goal so the non-interpretable would be the better move in that instance
 - If a bank uses it, making mistakes may cost some money, but non-interpretability may cause a lawsuit
- Attribute Selection
 - Select a subset of attributes that are relevant for the target data mining task
 - Maximize accuracy of the model, and minimize the number of selected attributes
 - Trade offs
 - Subset A is more accurate than B, but B has fewer attributes

Three Approaches for Coping with Multi Objective Problems

- Transforming multi-objective to single-objective
 - Assigning a weight to each objective and combining values of weighted criteria into single value by adding or multiplying all the weighted criteria
 - $Q = w_1 * c_1 + w_2 * c_2 + \dots + w_n * c_n$ Or
 - $Q = c_1^{w_1} * c_2^{w_2} * \dots * c_n^{w_n}$
 - A couple scenarios
 - Rule induction algorithms for classification
 - Attribute selection algorithms for classification
 - Arguments For
 - Simplicity
 - Easy to use
 - Arguments Against
 - Setting of weights is ad-hoc
 - Each weight is "magic", justified vaguely
 - Hard for user to define the best setting of weights a priori without knowing results of the research
 - But you could basically brute force it and pick the best weights
 - Mixing different units of measurement
 - Model quality criterias have different scales in their units of measurement
 - Can be fixed with normalization but only to a certain extent
 - Mixing apples/oranges
 - Non commensurable criteria should not be added together
 - For example 50k salary + 5 dependents
- Lexicographic Approach
 - Assign different priorities to different objectives then optimize each in order of priority
 - Usually compare performance measure for the highest priority objective unless one is significantly better than the other
 - Arguments For
 - Recognizing non commensurability of different quality data
 - Avoids the 3 drawbacks of weighted formula approach
 - Simpler and easier than Pareto
 - Arguments Against
 - Introducing a new Ad-Hoc parameter
 - Requires one to specify a tolerance threshold for each criterion
- Pareto Approach
 - Use multi objective algorithm to solve the original multi objective problem
 - Pareto Dominance
 - s1 Dominates s2 iff s1 is strictly better than s2 w.r.t. at least one of the objectives being optimized and s1 is not worse than s2 w.r.t. all criteria being optimized

- there's one c_i s.t. $s_1(c_i) > s_2(c_i)$ and
- for all c_i , $i=1\dots k$, $s_1(c_i) \geq s_2(c_i)$
- See figure 1 of [paper](#)
- Pareto vs Single Objective
 - Multi objective returns a set of non dominated solutions rather than a single solution
 - Search by multi objective should explore wider area of the search space and track non dominated solutions found to find as many solutions as possible
 - Much more complex than single objective
- Arguments For
 - Multiple runs of a single-objective optimization algorithm is inefficient and ineffective
 - Minimum description length principle comes with a price
 - How to encode hypothesis and its data exceptions into bits of info
- Arguments Against
 - Multiple runs of a single-objective optimization algorithm seems "enough"
 - Minimum description length principle seems "enough"
- Moral of the story: Pareto and Lexicographic are the way to go for multi objective

MLSS

2022-03-21

Lead Scribe Chamudi

The Indian Buffet Process: An Introduction and Review

Presented by Surbhi

The Indian buffet process is a stochastic process defining a probability distribution over equivalence classes of sparse binary matrices with a finite number of rows and infinite number of columns.

unsupervised learning aims to recover the latent structure responsible for generating observed data

key problems faced by unsupervised learning

- determining the amount of latent structure (no. of clusters, dimensions or variables)

Problem assumes that there is a single, finite dimensional representation that correctly characterizes the properties of the observed object.

An alternative is to assume that the amount of latent structure is potentially unbounded, and the observed objects only manifest a sparse subset of those classes or features.

This paper

- summarizes the extension of nonparametric approach to models in which objects are presented using an unknown number of latent features.
- how the Indian buffet process can be used to specify prior distributions in latent feature models, using a simple linear-Gaussian model to show how such models can be defined and used.

Latent Class Models

LCM relates a set of observed (Usually discrete) multivariate variables to a set of latent variables

Finite Mixture Model

A finite mixture model (FMM) is a statistical model that assumes the presence of unobserved groups, called latent classes, within an overall population. Each latent class can be fit with its own regression model, which may have a linear or generalized linear response function.

assumes that the assignment of an object to a class is independent of the assignments of all other objects.

if there are K classes

$$P(\mathbf{c}|\theta) = \prod_{i=1}^N P(c_i|\theta) = \prod_{i=1}^N \theta_{c_i},$$

Infinite Mixture Models

defining an infinite mixture model means that we want to specify the probability of \mathbf{X} in terms of infinitely many classes, modifying Equation 1 to become

$$p(\mathbf{X}|\theta) = \prod_{i=1}^N \sum_{k=1}^{\infty} p(\mathbf{x}_i|c_i = k) \theta_k,$$

The Chinese Restaurant Process

Named based upon a metaphor

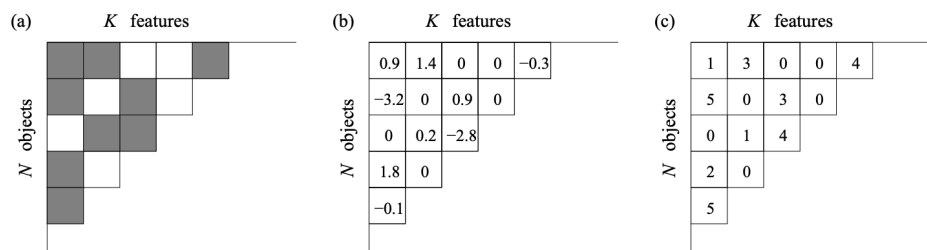
The objects are customers in a restaurant, and the classes are the tables at which they sit.

If we assume an ordering on our N objects, then we can assign them to classes sequentially using the method specified by the CRP, letting objects play the role of customers and classes play the role of tables. The i th object would be assigned to the k th class with probability

Latent Feature Models

Each object is represented by a vector of latent feature values \mathbf{f}_i , and the properties \mathbf{x}_i are generated from a distribution determined by those latent feature values.

Latent feature values can be continuous, as in factor analysis and probabilistic principal component analysis or discrete, as in cooperative vector quantization.



Feature matrices. A binary matrix Z , as shown in **a**, can be used as the basis for sparse infinite latent feature models, indicating which features take non-zero values. Elementwise multiplication of Z by a matrix V of continuous values gives a representation like that shown in **b**. If V contains discrete values, we obtain a representation like that shown in **c**.

The Indian Buffet Process

Define a distribution over infinite binary matrices by specifying a procedure by which customers (objects) choose dishes (features).

Indian buffet process (IBP), N customers enter a restaurant one after another.

Each customer encounters a buffet consisting of infinitely many dishes arranged in a line.

The first customer starts at the left of the buffet and takes a serving from each dish, stopping after Poisson(α) number of dishes as his plate becomes overburdened.

The i th customer moves along the buffet, sampling dishes in proportion to their popularity, serving himself with probability m_k/i , where m_k is the number of previous customers who have sampled a dish. Having reached the end of all previous sampled dishes, the i th customer then tries a Poisson(α/i) number of new dishes.

We can indicate which customers chose which dishes using a binary matrix Z with N rows and infinitely many columns, where $z_{ik} = 1$ if the i th customer sampled the k th dish.

Auto-Encoding Variational Bayes

Presented by Chan

Generative model

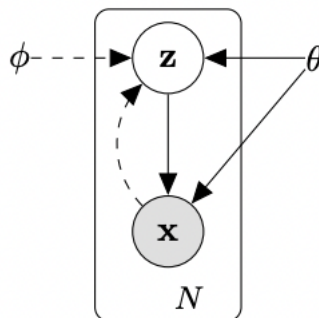
ML model that learns a given learning data and generates similar data according to the distribution of train data

Explicit density

Tractable density and approximate density

Normally requires high mathematical knowledge to understand

Manifold Hypothesis



The type of directed graphical model under consideration. Solid lines denote the generative model $p_{\theta}(z)p_{\theta}(x|z)$, dashed lines denote the variational approximation $q_{\phi}(z|x)$ to the intractable posterior $p_{\theta}(z|x)$. The variational parameters ϕ are learned jointly with the generative model parameters θ .

Methodology

Maximum likelihood : $\arg \max_{\theta} [p_{\theta}(x) = \int z p_{\theta}(x, z) = \int z p_{\theta}(x|z) p_{\theta}(z)]$

Variation Inference

$$q_{\phi}(z|x) \approx p_{\theta}(z|x)$$

The variational bound

$$\int z q(z|x) = 1$$

$$p(x) = p(z, x) / p(z|x)$$

$$\mathbb{E}[X] = \int x f(x)$$

$$\text{DKL}(P||Q) = \int p(x) \log(p(x) / q(x))$$

$$\log p_{\theta}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log q_{\phi}(\mathbf{z}|\mathbf{x}) + \log p_{\theta}(\mathbf{x}, \mathbf{z})]$$

Evidence Lower Bound

$$\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [-\log q_{\phi}(\mathbf{z}|\mathbf{x}) + \log p_{\theta}(\mathbf{x}, \mathbf{z})]$$

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})} [\log p_{\theta}(\mathbf{x}^{(i)}|\mathbf{z})]$$

Stochastic Gradient Variational Bayes(SGVB)

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\phi}(\mathbf{z})} [f(\mathbf{z}) \nabla_{q_{\phi}(\mathbf{z})} \log q_{\phi}(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^L f(\mathbf{z}^{(l)}) \nabla_{q_{\phi}(\mathbf{z}^{(l)})} \log q_{\phi}(\mathbf{z}^{(l)})$$

where $\mathbf{z}^{(l)} \sim q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})$

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})} [\log p_{\theta}(\mathbf{x}^{(i)}|\mathbf{z})]$$

$$\simeq \left[-D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z})) + \frac{1}{L} \sum_{l=1}^L \log p_{\theta}(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) \right]$$

when both the prior $p_0(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ and the posterior approximation $q_{\phi}(\mathbf{z}|\mathbf{x})$ are Gaussian and j is the dimensionality of \mathbf{z}

$$\begin{aligned} -D_{KL}(q_{\phi}(\mathbf{z})||p_{\theta}(\mathbf{z})) &= \int q_{\theta}(\mathbf{z}) (\log p_{\theta}(\mathbf{z}) - \log q_{\theta}(\mathbf{z})) d\mathbf{z} \\ &= \frac{1}{2} \sum_{j=1}^J (1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2) \end{aligned}$$

Reparameterization Tricks

$\mathbf{z}^{(l)} \sim q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})$ cannot differentiate

Solution

$\mathbf{z} = \mu + \sigma\epsilon$, where ϵ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$.

MLSS

2022-03-28

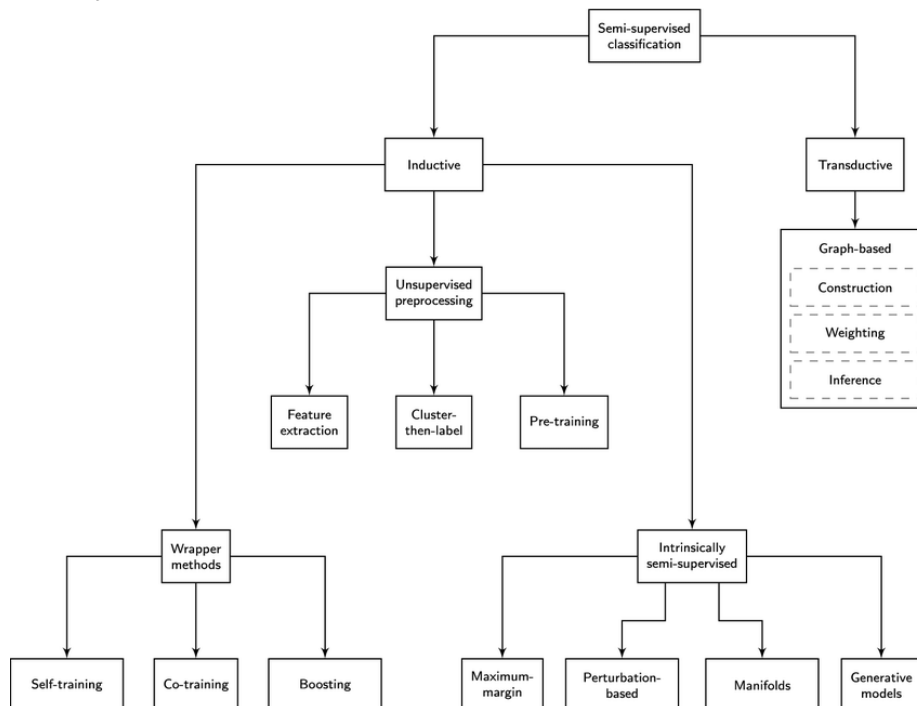
Lead Scribe - Chan

A Survey on Semi-Supervised Learning

Presented by Derek

- Semi-supervised learning
- In this survey
 - attempting to give a comprehensive overview of the state of semi-supervised learning
 - new taxonomy for semi-supervised classification methods
- Background
- Assumptions SSL
 - Underlying marginal data distribution ($p(\mathbf{x})$) over the input space contains info about the posterior distribution $p(\mathbf{y}|\mathbf{x})$
 - Multiple assumptions based on interaction of $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$
 - assumption \rightarrow how does $p(\mathbf{x})$ tells us something about $p(\mathbf{y}|\mathbf{x})$?

- not time-series data
- points are not related to each other
- samples are not conditional to each other.
- classes are separable
- Smoothness
 - For two input points x, x' that are close in the input space, the labels y, y' should be the same
- Low-Density
 - implies the decision boundary of a classifier should pass through low density regions in the input space
 - where few points are observed
 - smoothness assumption is not violated
 - but if boundary is in high density, could violate the smoothness assumption
- Manifold
 - manifold hypothesis - data lives in a lower dimension than we expect.
- Connection to Clustering
 - cluster assumption is often included
- When does SSL work?
 - algorithm needs to be able to extract the info relating $p(x)$ to $p(y|x)$ [hard to answer]
- Empirical Eval of SSL
 - Many decisions influence the relative performance of algorithms
 - Data set partitioning, hyperparameter tuning
 - With SSL we also have
 - what should be labeled and what should not be
 - can choose to eval performance on unlabeled data or on a disjoint set.
 - Toy data sets are usually used to show viability of a new approach
 - Data set choice can have large impact on performance.
- Taxonomy



- Inductive Methods
 - wrapper methods
 - train classifiers on labeled data → use predictions to generate additional labeled data - > retrain on the pseudo-labeled data in addition to existing labeled data.
 - among the oldest and most widely known algorithms for SSL
 - Advantage: can be used with ~any supervised base learner
 - Broken into self training, co-training, and pseudo-labelled boosting methods.
 - self training

- most basic pseudo labelling approach
- single supervised classifier iteratively trained on both labelled data and data that's been pseudo labelled
- Many design decisions
 - Selection of data to pseudo label
 - Re-use of pseudo labelled data
 - Stopping criteria
- co-training
 - 2 or more supervised classifiers that are iteratively trained on the labelled data, adding most confident predictions to the labeled set of the other supervised classifier at each iteration.
 - important that base learners are not strongly correlated in their predictions (classifier diversity)
 - single view and multi view
 - co-regularization
- Boosting
 - bagging
 - each base learner is given a set of i data points, which are sampled uniformly at random with replacement from the original data set
 - boosting
 - each base learner is dependent on previous base learners
 - SSMBoost
 - AdaBoost extended to SSL setting
 - ASSEMBLE
 - gets rid of base learners
 - SemiBoost
 - Relies on the manifold assumption, using principles from graph-based methods
 - Other SSL Boosting Methods
 - RegBoost
- unsupervised preprocessing
 - Uses unlabelled and labelled data in two separate stages
 - Feature extraction
 - Attempts to find a transformation of input data such that performance of the classifier improves or such that its construction becomes computationally more efficient
 - Many feature extraction methods operate without supervision (like PCA), others operate on labelled data and extract features with high predictive power
 - Relates to autoencoding (like we saw last week)

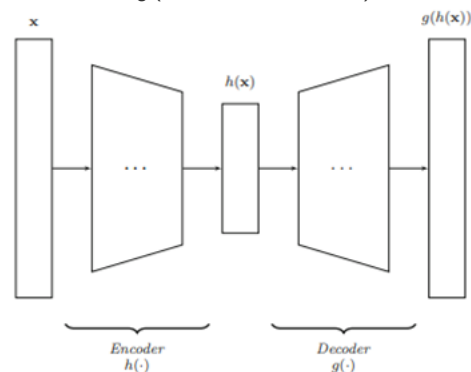
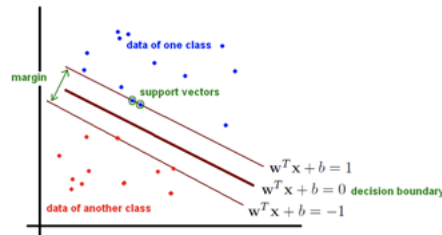


Fig. 4 Simplified representation of an autoencoder. The rectangles correspond to layers within the network; the trapeziums represent the encoder and decoder portions of the network, which can consist of multiple layers

- Cluster-then-label
 - Form a group of methods that join clustering and classification processes
 - First apply an unsupervised or semi-supervised clustering algorithm to all data, then uses resulting clusters to guide classification
- Pre-Training

- Unlabelled data used to guide decision boundary towards potentially interesting regions before applying supervised learning
 - intrinsically semi-supervised methods
 - Don't rely on intermediate steps or supervised base learners
 - Extensions of existing supervised methods to include unlabeled samples in the objective function

- Maximum-margin methods



- SVMs
 - Semi Supervised SVMs
 - Gaussian Processes
 - Density Regularization
 - Pseudo-labeling as a form of margin maximization
 - Perturbation-based methods (PB)
 - when we perturb a data point with some noise, the predictions for noisy and clean inputs should be similar
 - smoothness assumption
 - Methods of incorporating smoothness assumption into a given learning algorithm
 - Apply noise to input data and incorporate the difference between clean and noisy into the loss function
 - Implicitly apply noise to data points by perturbing the classifier
 - Semi-Supervised Neural Networks
 - Simplicity and efficiency of back propagation makes it easy to add an unsupervised component to the loss function
 - Because NNs are hierarchical, they are also viable candidates for semi-supervised approaches
 - Ladder networks
 - Extends a feedforward network to incorporate unlabelled data by using the feedforward part of the network as the encoder of a denoising autoencoder
 - Adding a decoder, and including a term in the cost function to penalize the reconstruction cost
 - Pseudo-ensembles
 - Perturbing the neural network model itself
 - Robustness through penalizing the differences between the activations of perturbed network and those of the original network for the same input
 - Unperturbed parent model is perturbed to obtain one or more child models
 - \square models
 - Computing perturbed models directly
 - Using dropout and penalizing differences in the final layer activation of the networks using squared loss
 - Mean teacher
 - Average weights = teacher
 - latest model = student
 - Temporal Ensembling
 - virtual adversarial training
 - semi-supervised mixup
- Manifolds
 - an m-dim manifold is a subspace of the original input
 - Manifold regularization
 - Introduce regularization term that captures the fact that manifolds represent lower dimension Euclidean space
 - Manifold approximation

- Manifold explicitly approximated, then used in a classification task
- Generative models
 - model that process the generation of data
 - Mixture models
 - When prior knowledge about $p(x,y)$ is available, GM can be strong
 - Data is composed of k Gaussian distributions → fix model as a mixture of k gaussian components
 - Generative adversarial networks
 - Variational autoencoders
- Transductive Methods
 - do not construct a classifier for entire input space
 - Limited to exactly those objects encountered during training phase.
 - info must be propagated through direct connections between data points (graph-based approaches)
 - cannot distinguish between training and testing phase
 - General Framework for graph based method
 - graph construction
 - creation
 - weighing
 - Most important aspect of graph-based methods
 - Form edges between nodes (adj. matrix)
 - attaching weights to them (weight matrix)
 - Adj. matrix construction
 - KNN, ϵ -neighborhood, b-matching
 - connects nodes to all nodes to which distance is at most ϵ
 - find subsets of edges in complete graph such that each node has degree b and the sum of edge weights is maxed.
 - Graph weighting
 - Simultaneous graph
 - Inference
 - forming predictions for the unlabelled data points
 - Hard label assignments: graph min-cut
 - Probabilistic label assignments: Markov random fields
 - Efficient Probabilistic label assignments: Gaussian random fields
 - like MRFs
 - Handling label noise and irregular graphs
 - approach that addresses two issues
 - handling label noise
 - influence of nodes with high degree in irregular graphs is large
 - Further research on graph based inference
 - absorption
 - class imbalance sensitivity
 - Scalable transductive learning
- Classification in network data
 - represented as graphs

Learning with Noisy Labels

Presented by Derek

- Intro
 - design supervised learning algo. learning from noisy labels
 - approaches
 - modified/Proxy loss is an unbiased estimate of the loss function
 - Based on the idea that the probability of misclassification under noisy distribution differs from clean distribution where a given threshold is used to decide the label.
- Problem Setup and Background
 - $\tilde{Y} \rightarrow$ incorrect data

- not every single label is wrong (<1)
- f is a decision function
- important quantities associated with f
- l is a convex function calibrated to a loss
- In the event f is not quantified in a minimization, the minimization is overall measurable functions.

• Methods of Unbiased Estimators

Lemma 1. Let $\ell(t, y)$ be any bounded loss function. Then, if we define,

$$\tilde{\ell}(t, y) := \frac{(1 - \rho_{-y}) \ell(t, y) - \rho_y \ell(t, -y)}{1 - \rho_{+1} - \rho_{-1}}$$

we have, for any t, y , $\mathbb{E}_{\tilde{y}} [\tilde{\ell}(t, \tilde{y})] = \ell(t, y)$.

Lemma 2. Let $\ell(t, y)$ be L -Lipschitz in t (for every y). Then, with probability at least $1 - \delta$,

$$\max_{f \in \mathcal{F}} |\hat{R}_{\tilde{\ell}}(f) - R_{\tilde{\ell}, D_{\rho}}(f)| \leq 2L_{\rho} \mathfrak{R}(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Theorem 3 (Main Result 1). With probability at least $1 - \delta$,

$$R_{\ell, D}(\hat{f}) \leq \min_{f \in \mathcal{F}} R_{\ell, D}(f) + 4L_{\rho} \mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}}.$$

- Furthermore, if ℓ is classification-calibrated, there exists a nondecreasing function ζ_{ℓ} with $\zeta_{\ell}(0) = 0$ such that,

$$R_D(\hat{f}) - R^* \leq \zeta_{\ell} \left(\min_{f \in \mathcal{F}} R_{\ell, D}(f) - \min_f R_{\ell, D}(f) + 4L_{\rho} \mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}} \right).$$

• Method of Label-Dependent Costs

Lemma 7. Denote $P(Y = 1|X)$ by $\eta(X)$ and $P(\tilde{Y} = 1|X)$ by $\tilde{\eta}(X)$. The Bayes classifier under the noisy distribution, $\tilde{f}^* = \operatorname{argmin}_f E_{(X, \tilde{Y}) \sim D_{\rho}} [1_{\{\operatorname{sign}(f(X)) \neq \tilde{Y}\}}]$ is given by,

$$\tilde{f}^*(x) = \operatorname{sign}(\tilde{\eta}(x) - 1/2) = \operatorname{sign} \left(\eta(x) - \frac{1/2 - \rho_{-1}}{1 - \rho_{+1} - \rho_{-1}} \right).$$

Lemma 8 (α -weighted Bayes optimal [Scott, 2012]). Define U_{α} -risk under distribution D as

$$R_{\alpha, D}(f) = E_{(X, Y) \sim D} [U_{\alpha}(f(X), Y)].$$

Then, $f_{\alpha}^*(x) = \operatorname{sign}(\eta(x) - \alpha)$ minimizes U_{α} -risk.

Theorem 9 (Main Result 2). For the choices,

$$\alpha^* = \frac{1 - \rho_{+1} + \rho_{-1}}{2} \text{ and } A_{\rho} = \frac{1 - \rho_{+1} - \rho_{-1}}{2},$$

there exists a constant B_X that is independent of f such that, for all functions f ,

$$R_{\alpha^*, D_{\rho}}(f) = A_{\rho} R_D(f) + B_X.$$

Corollary 10. The α^* -weighted Bayes optimal classifier under noisy distribution coincides with that of 0-1 loss under clean distribution:

$$\operatorname{argmin}_f R_{\alpha^*, D_{\rho}}(f) = \operatorname{argmin}_f R_D(f) = \operatorname{sign}(\eta(x) - 1/2).$$

- Proposed Proxy Surrogate Losses
- Given a surrogate loss function and this decomposition

$$\ell(t, y) = 1_{\{y=1\}} \ell_1(t) + 1_{\{y=-1\}} \ell_{-1}(t)$$

Theorem 11 (Main Result 3). Consider the empirical risk minimization problem with noisy labels:

$$\hat{f}_{\alpha} = \operatorname{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell_{\alpha}(f(X_i), \tilde{Y}_i).$$

Define ℓ_{α} as an α -weighted margin loss function of the form:

$$\ell_{\alpha}(t, y) = (1 - \alpha) 1_{\{y=1\}} \ell(t) + \alpha 1_{\{y=-1\}} \ell(-t) \quad (1)$$

- where $\ell : \mathbb{R} \rightarrow [0, \infty)$ is a convex loss function with Lipschitz constant L such that it is classification-calibrated (i.e. $\ell'(0) < 0$). Then, for the choices α^* and A_{ρ} in Theorem 9, there exists a nondecreasing function $\zeta_{\ell_{\alpha^*}}$ with $\zeta_{\ell_{\alpha^*}}(0) = 0$, such that the following bound holds with probability at least $1 - \delta$:

$$R_D(\hat{f}_{\alpha^*}) - R^* \leq A_{\rho}^{-1} \zeta_{\ell_{\alpha^*}} \left(\min_{f \in \mathcal{F}} R_{\alpha^*, D_{\rho}}(f) - \min_f R_{\alpha^*, D_{\rho}}(f) + 4L \mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2n}} \right).$$

• Conclusions

- Proposed methods are competitive and able to tolerate moderate-high noise
- Methods can benefit from knowing the noise rates
- Algorithms give a new family of methods applicable to positive unbalanced learning problems

Preparing to present a paper

Questions that will help organize your preparation, but may apply variably to different readings:

- What is the key question that drove the research?
- What is the main finding?
- What is the model assumed in the paper?
- Did they include experimental results? If so:
 - do the experiments support the claims?
 - what additional experiments would help make the result make more sense?
 - how broad are the experiments, are the context-specific or general?
- Is there an analytical result? if so:
 - do the conditions for the proof make sense?
 - are they realistic?
 - what questions do you have about the proof?

You may plan to use slides if that make you more comfortable or you can show the paper itself. You may also show other materials if appropriate and you can seed the day's notes.

Posting Notes

First time:

1. Go to the notes page of the repository
2. Click add a file, choose create a new file
3. Add your notes
4. At the bottom choose "propose changes"
5. If applicable, navigate to your fork, to the branch you made to add additional files (eg images)
6. Open a pull request from your fork/branch to the the course repo/main.

Tip

These are a rough outline, if you need help, definitely feel free to ask.
Once you do it, feel free to add more detail.