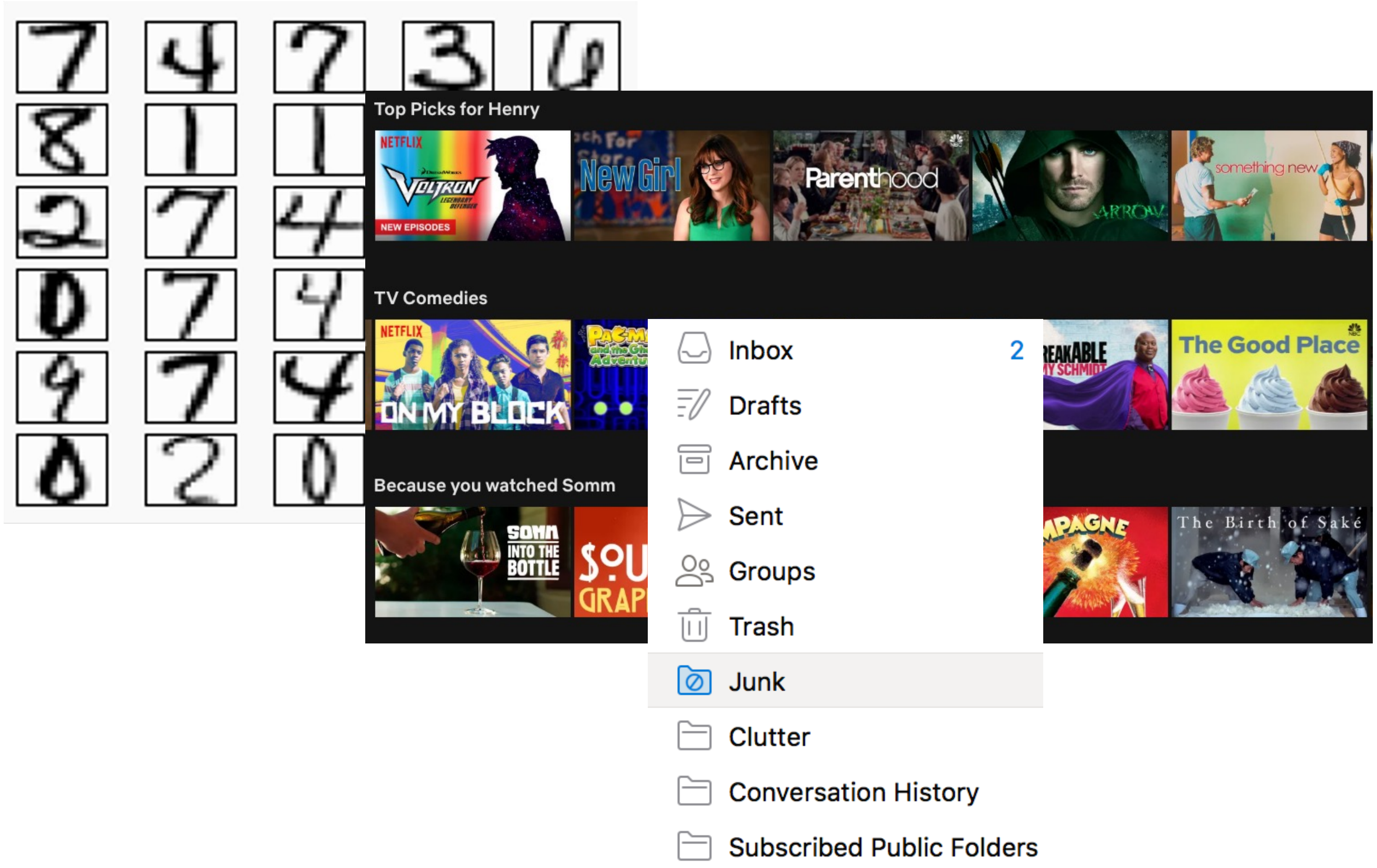
10-701: Introduction to Machine Learning Lecture 1 – Problem Formulation & Notation

Henry Chai & Zack Lipton

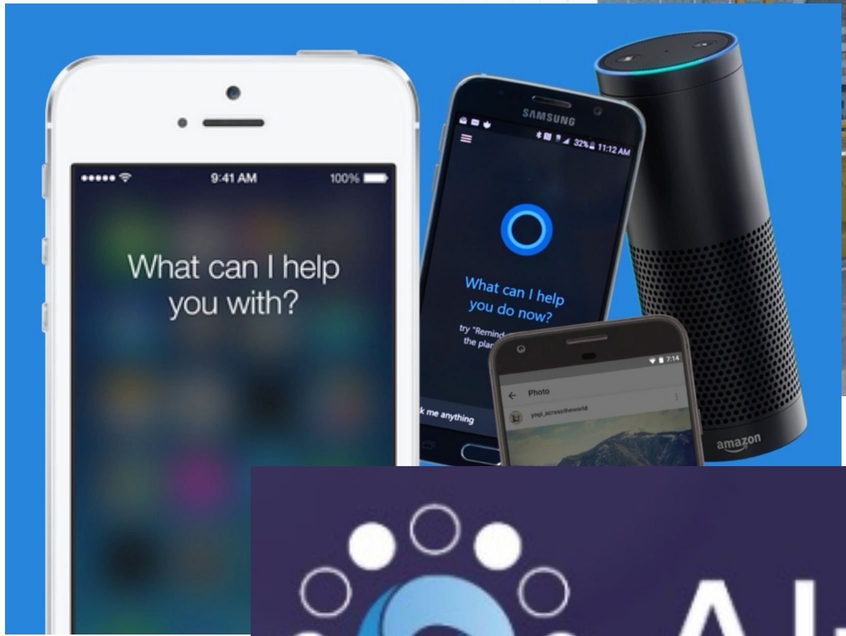
8/28/23

What is Machine Learning?

Machine Learning (A long long time ago…)



Machine Learning (A short time ago…)



AlphaGo

Machine Learning (Now)

Machine Learning (Now)

What is ~~Machine Learning~~ 10-701? (A short time ago…)

- Supervised Models
  - Decision Trees
  - KNN
  - Naïve Bayes
  - Perceptron
  - Logistic Regression
  - Linear Regression
  - Neural Networks
  - SVMs
- Unsupervised Models
  - K-means
  - PCA
- Ensemble Methods
- Graphical Models
  - Bayesian Networks
  - HMMs
- Learning Theory
- Reinforcement Learning
- Important Concepts
  - Feature Engineering
  - Regularization and Overfitting
  - Experimental Design

What is ~~Machine Learning~~ 10-701? (Now)

- Supervised Models
  - Decision Trees
  - KNN
  - Naïve Bayes
  - Perceptron
  - Logistic Regression
  - Linear Regression
  - Neural Networks
  - SVMs
- Unsupervised Models
  - K-means
  - PCA
- Ensemble Methods
- Graphical Models
  - Bayesian Networks
  - HMMs
- Learning Theory
- Reinforcement Learning
- Important Concepts
  - Feature Engineering
  - Regularization and Overfitting
  - Experimental Design

Deep Learning & Generative AI

Defining a Machine Learning Task (Mitchell, 97)

- A computer program **learns** if its *performance*, *P*, at some *task*, *T*, improves with *experience*, *E*.
- Three components
  - Task, T
  - Performance metric, P
  - Experience, E

Defining a Machine Learning Task: Example

- Learning to approve loans/lines of credit

- Three components
  - Task, T
  - Performance metric, P
  - Experience, E

Defining a Machine Learning Task: Example

- Learning to approve loans/lines of credit
- Three components
  - Task, T
  - Performance metric, P
  - Experience, E

Things Machine Learning Isn't

- Neutral?
  - Do you agree or disagree with the following statement: "Because machine learning uses algorithms, math, and data, it is inherently neutral or impartial?"

Defining a Machine Learning Task: Example

- Learning to

- Three components
  - Task, T
  - Performance metric, P
  - Experience, E

Defining a Machine Learning Task: Example

- Learning to
- Three components
  - Task, T
  - Performance metric, P
  - Experience, E

Our first Machine Learning Task

- Learning to diagnose heart disease

  as a **(supervised) binary classification task**

features labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

Our first Machine Learning Task

- Learning to diagnose heart disease

  as a **(<u>supervised</u>) binary classification task**

features | labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

Our first Machine Learning Task

- Learning to diagnose heart disease

  as a **(supervised)** **<u>binary classification</u> task**

features | labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

Our first Machine Learning Task

- Learning to diagnose heart disease

  as a **(supervised)** **<u>classification</u> task**

features labels

data points

Family History	Resting Blood Pressure	Cholesterol	Risk
Yes	Low	Normal	Low Risk
No	Medium	Normal	Low Risk
No	Low	Abnormal	Medium Risk
Yes	Medium	Normal	High Risk
Yes	High	Abnormal	High Risk

Our first Machine Learning Task

- Learning to diagnose heart disease

  as a **(supervised)** **<u>regression</u> task**

features targets

data points

Family History	Resting Blood Pressure	Cholesterol	Medical Costs
Yes	Low	Normal	$0
No	Medium	Normal	$20
No	Low	Abnormal	$30
Yes	Medium	Normal	$100
Yes	High	Abnormal	$5000

Our first Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the dataset

features

labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

Is this a "good" Classifier?

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the dataset

features labels

data points

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

Training vs. Testing

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the **training** dataset (Yes)

training dataset

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

Training vs. Testing

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the **training** dataset (Yes)
- A **test** dataset is used to evaluate a classifier's **predictions**

test dataset

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	**Predictions**
No	Low	Normal	No	Yes
No	High	Abnormal	Yes	Yes
Yes	Medium	Abnormal	Yes	Yes

- The **error rate** is the proportion of data points where the prediction is wrong

Training vs. Testing

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the **training** dataset (Yes)
- A **test** dataset is used to evaluate a classifier's **predictions**

test dataset

Family History	Resting Blood Pressure	Cholesterol	Heart Disease?	**Predictions**
No	Low	Normal	No	Yes
No	High	Abnormal	Yes	Yes
Yes	Medium	Abnormal	Yes	Yes

- The **test error rate** is the proportion of data points in the test dataset where the prediction is wrong $(1/3)$

A Typical (Supervised) Machine Learning Routine

- Step 1 – training
  - Input: a labelled training dataset
  - Output: a classifier
- Step 2 – testing
  - Inputs: a classifier, a test dataset
  - Output: predictions for each test data point
- Step 3 – evaluation
  - Inputs: predictions from step 2, test dataset labels
  - Output: some measure of how good the predictions are; usually (but not always) error rate

Our first Machine Learning Classifier

- A **classifier** is a function that takes feature values as input and outputs a label
- Majority vote classifier: always predict the most common label in the training dataset

labels

data points

Heart Disease?	Predictions
No	Yes
No	Yes
Yes	Yes
Yes	Yes
Yes	Yes

- The training error rate is $2/5$

Notation

- Feature space, $\mathcal{X}$
- Label space, $\mathcal{Y}$
- (Unknown) Target function, $c^*: \mathcal{X} \rightarrow \mathcal{Y}$
- Training dataset:

$$\mathcal{D} = \left\{\left(\boldsymbol{x}^{(1)}, c^*\left(\boldsymbol{x}^{(1)}\right) = y^{(1)}\right), \left(\boldsymbol{x}^{(2)}, y^{(2)}\right) \dots, \left(\boldsymbol{x}^{(N)}, y^{(N)}\right)\right\}$$

- Data point:

$$\left(\boldsymbol{x}^{(n)}, y^{(n)}\right) = \left(x_1^{(n)}, x_2^{(n)}, \dots, x_D^{(n)}, y^{(n)}\right)$$

- Classifier, $h : \mathcal{X} \rightarrow \mathcal{Y}$
- Goal: find a classifier, h, that best approximates c^*

Evaluation

- Loss function, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
  - Defines how "bad" predictions, $\hat{y} = h(\boldsymbol{x})$, are compared to the true labels, $y = c^*(\boldsymbol{x})$
  - Common choices

1. Squared loss (for regression): $\ell(y, \hat{y}) = (y - \hat{y})^2$
2. Binary or 0-1 loss (for classification):

$$\ell(y, \hat{y}) = \mathbb{1}(y \neq \hat{y})$$

- Error rate:

Notation: Example

- Majority vote classifier: always predict the most common label in the **training** dataset

	x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?	$\hat{y}$ Predictions
	Yes	Low	Normal	No	No
$\boldsymbol{x}^{(2)}$	No	Medium	Normal	No	No
	No	Low	Abnormal	Yes	Yes
	Yes	Medium	Normal	Yes	Yes
	Yes	High	Abnormal	Yes	Yes

- $N = 5$ and $D = 3$
- $\boldsymbol{x}^{(2)} = \left(x_1^{(2)} = \text{"No"}, x_2^{(2)} = \text{"Medium"}, x_3^{(2)} = \text{"Normal"}\right)$

Our second Machine Learning Classifier

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump: based on a single feature, x_d, predict the most common label in the training dataset among all data points that have the same value for x_d

Our second Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x'_1, \dots, x'_D) = \begin{cases} ??? & \boxed{\text{if } x'_1 = \text{"Yes"}} \\ ??? & \text{otherwise} \end{cases}$$

Our second Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x'_1, \dots, x'_D) = \begin{cases} \text{“Yes” if } x'_1 = \text{“Yes”} \\ ??? \ \boxed{\text{otherwise}} \end{cases}$$

Our second Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?
Yes	Low	Normal	No
No	Medium	Normal	No
No	Low	Abnormal	Yes
Yes	Medium	Normal	Yes
Yes	High	Abnormal	Yes

- Decision stump on x_1:

$$h(\boldsymbol{x}') = h(x'_1, \dots, x'_D) = \begin{cases} \text{"Yes" if } x'_1 = \text{"Yes"} \\ \text{"No" otherwise} \end{cases}$$

Our second Machine Learning Classifier: Example

- Alright, let's actually (try to) extract a pattern from the data

x_1 Family History	x_2 Resting Blood Pressure	x_3 Cholesterol	y Heart Disease?	$\hat{y}$ Predictions
Yes	Low	Normal	No	Yes
No	Medium	Normal	No	No
No	Low	Abnormal	Yes	No
Yes	Medium	Normal	Yes	Yes
Yes	High	Abnormal	Yes	Yes

Decision Stumps: Questions

1. How can we pick which feature to split on?

2. Why stop at just one feature?

Key Takeaways

- Components of a machine learning problem
- Algorithmic bias
- Components of a labelled dataset for supervised learning
- Training vs. test datasets
- Majority vote classifier
- Decision stumps

Logistics: Course Website

https://machinelearningcmu.github.io/F23-10701/

Logistics: Course Syllabus

https://machinelearningcmu.github.io/F23-10701/#Syllabus

- This whole section is **required** reading

Logistics: Grading

https://machinelearningcmu.github.io/F23-10701/#Syllabus

- 25% midterm
- 25% final
- 24% homework assignments
  - 4 assignments at 6% each
- 26% project
  - You must work on the project in groups of 3 or 4

Logistics: Late Policy

https://machinelearningcmu.github.io/F23-10701/#Syllabus

- 4 grace days for use across all homework assignments
- Only 2 grace days may be used per homework
- Late submissions w/o grace days:
  - 1 day late = 50% multiplicative penalty
  - 2 days late = 25% multiplicative penalty
- No submissions accepted more than 3 days late
- Grace days cannot be applied to project deliverables

Logistics: Collaboration Policy

https://machinelearningcmu.github.io/F23-10701/#Syllabus

- Collaboration on homework assignments is encouraged but must be documented
- **You must always write your own code/answers**
  - You may not re-use code/previous versions of the homework, whether your own or otherwise
- Good approach to collaborating on programming assignments:
  1. Collectively sketch pseudocode on an impermanent surface, then
  2. Disperse, erase all notes and start from scratch

Logistics: Technologies

https://machinelearningcmu.github.io/F23-10701/#Syllabus

- Piazza, for course discussion: https://piazza.com/class/llkvlxou7zs3pz
- Gradescope, for submitting homework assignments: https://www.gradescope.com/courses/580643
- Panopto, for lecture recordings: https://scs.hosted.panopto.com/Panopto/Pages/Sessions/List.aspx?folderID=d9d7c7cf-d714-490d-a9e6-b06600f67388

Logistics: Lecture Schedule

https://machinelearningcmu.github.io/F23-10701/#Schedule

Schedule

Date	Topic	Slides	Readings/Resources
M, Aug-28	Introduction: Logistics, Notation & Problem Formulation		
W, Aug-30	Decision Trees		
M, Sep-4	Labor Day - No Class		
W, Sep-6	KNNs & Model Selection		
M, Sep-11	Linear Regression		
W, Sep-13	Regularization		
M, Sep-18	MLE/MAP		
W, Sep-20	Naïve Bayes		

Logistics: Exam Schedule

https://machinelearningcmu.github.io/F23-10701/#Schedule

Schedule

Date	Topic	Slides	Readings/Resources
⋮			
M, Oct-30	Unsupervised Learning & Dimensionality Reduction		
Tu, Oct-31	Midterm Exam (Evening)		
⋮			
W, Dec-6	Privacy		
TBD, TBD	Final Exam (Registrar Scheduled)		

Logistics: Programming Assignments

https://machinelearningcmu.github.io/F23-10701/#Assignments

Assignments

Release Date	Topic	Files	Due Date
Sep-6	HW1: Decision Trees & KNNs	(Not released yet)	Sep-20
Sep-20	HW2: Linear Regression & Naïve Bayes	(Not released yet)	Oct-4
Oct-4	HW3: Bayesian Networks & Reinforcement Learning	(Not released yet)	Oct-11
Oct-11	HW4: Feed-forward Neural Networks	(Not released yet)	Oct-25

Logistics: Office Hours

https://machinelearningcmu.github.io/F23-10701/#Calendar