Course Introduction
How to define a structured prediction problem

STRUCTURED PREDICTION
Structured vs. Unstructured Data

**Structured Data Examples**
- database entries
- transactional information
- wikipedia infobox
- knowledge graphs
- hierarchies

**Unstructured Data Examples**
- written text
- images
- videos
- spoken language
- music
- sensor data
Structured vs. Unstructured Data

Select all that apply:
Which of the following are structured data?
- spreadsheet
- XML data
- JSON data
- mathematical equations

Answer:
Structured Prediction

- Most of the models we’ve seen so far were for **classification**
  - Given observations: \( x = (x_1, x_2, \ldots, x_K) \)
  - Predict a (binary) **label**: \( y \)

- Many real-world problems require **structured prediction**
  - Given observations: \( x = (x_1, x_2, \ldots, x_K) \)
  - Predict a **structure**: \( y = (y_1, y_2, \ldots, y_J) \)

- Some **classification** problems benefit from **latent structure**
Structured Prediction

Classification / Regression
1. Input can be semi-structured data
2. Output is a single number (integer / real)
3. In linear models, features can be arbitrary combinations of [input, output] pair
4. Output space is small
5. Inference is trivial

Structured Prediction
1. Input can be semi-structured data
2. Output is a sequence of numbers representing a structure
3. In linear models, features can be arbitrary combinations of [input, output] pair
4. Output space may be exponentially large in the input space
5. Inference problems are NP-hard or #P-hard in general and often require approximations
Structured Prediction Examples

• **Examples of structured prediction**
  – Part-of-speech (POS) tagging
  – Handwriting recognition
  – Speech recognition
  – Object detection
  – Scene understanding
  – Machine translation
  – Protein sequencing
**Part-of-Speech (POS) Tagging**

<table>
<thead>
<tr>
<th>Sample 1:</th>
<th>n</th>
<th>v</th>
<th>p</th>
<th>d</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>arrow</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample 2:</th>
<th>n</th>
<th>n</th>
<th>v</th>
<th>d</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>arrow</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample 3:</th>
<th>n</th>
<th>v</th>
<th>p</th>
<th>n</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>flies</td>
<td>fly</td>
<td>with</td>
<td>their</td>
<td>wings</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample 4:</th>
<th>p</th>
<th>n</th>
<th>n</th>
<th>v</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>with</td>
<td>time</td>
<td>you</td>
<td>will</td>
<td>see</td>
<td></td>
</tr>
</tbody>
</table>
Dataset for Supervised Part-of-Speech (POS) Tagging

Data: $\mathcal{D} = \{x^{(n)}, y^{(n)}\}_{n=1}^{N}$

<table>
<thead>
<tr>
<th>Sample 1:</th>
<th>n</th>
<th>v</th>
<th>p</th>
<th>d</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>arrow</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample 2:</th>
<th>n</th>
<th>n</th>
<th>v</th>
<th>d</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>arrow</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample 3:</th>
<th>n</th>
<th>v</th>
<th>p</th>
<th>n</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>flies</td>
<td>fly</td>
<td>with</td>
<td>their</td>
<td>wings</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample 4:</th>
<th>p</th>
<th>n</th>
<th>n</th>
<th>v</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>with</td>
<td>time</td>
<td>you</td>
<td>will</td>
<td>see</td>
<td></td>
</tr>
</tbody>
</table>
Handwriting Recognition

Sample 1:

unexpected

Sample 2:

volcanic

Sample 2:

embraces

Figures from (Chatzis & Demiris, 2013)
Dataset for Supervised Handwriting Recognition

Data: \( \mathcal{D} = \{ x^{(n)}, y^{(n)} \}_{n=1}^{N} \)

<table>
<thead>
<tr>
<th>Sample 1:</th>
<th>( y^{(1)} )</th>
<th>( x^{(1)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 2:</td>
<td>( y^{(2)} )</td>
<td>( x^{(2)} )</td>
</tr>
<tr>
<td>Sample 3:</td>
<td>( y^{(3)} )</td>
<td>( x^{(3)} )</td>
</tr>
</tbody>
</table>

Figures from (Chatzis & Demiris, 2013)
Dataset for Supervised Phoneme (Speech) Recognition

Data: \( \mathcal{D} = \{ x^{(n)}, y^{(n)} \}_{n=1}^{N} \)

Sample 1:

\[ \text{Sample 1: } \{ \text{h#, dh, ih, s, w, uh, z, iy, z, iy} \} \]

Sample 2:

\[ \text{Sample 2: } \{ \text{f, ao, r, ah, s, s, h#} \} \]

Figures from (Jansen & Niyogi, 2013)
Case Study: Object Recognition

Data consists of images $x$ and labels $y$. 

- $x^{(1)}$: pigeon
- $y^{(1)}$
- $x^{(2)}$: rhinoceros
- $y^{(2)}$
- $x^{(3)}$: leopard
- $y^{(3)}$
- $x^{(4)}$: llama
- $y^{(4)}$
Case Study: Object Recognition

Data consists of images \( x \) and labels \( y \).

- Preprocess data into “patches”
- Posit a latent labeling \( z \) describing the object’s parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- \( z \) is not observed at train or test time
Case Study: Object Recognition

Data consists of images $x$ and labels $y$.

- Preprocess data into “patches”
- Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- $z$ is not observed at train or test time
Case Study: Object Recognition

Data consists of images $x$ and labels $y$.

- Preprocess data into “patches”
- Posit a latent labeling $z$ describing the object’s parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- $z$ is not observed at train or test time
Structured Prediction

Preview of challenges to come...

• Consider the task of finding the **most probable assignment** to the output

  \[
  \hat{y} = \arg\max_y p(y|x)
  \]

  where \( y \in \{+1, -1\} \)

  \[
  \hat{v} = \arg\max_v p(v|t)
  \]

  where \( v \in Y \)

  and \(|Y|\) is very large
Structured Prediction

Data

Inference

Model

Objective

Learning

(Inference is usually called as a subroutine in learning)
Structured Prediction

The **data** inspires the structures we want to predict.

Our **model** defines a score for each structure.

It also tells us what to optimize.

**Inference** finds \{best structure, marginals, partition function\} for a new observation.

(Inference is usually called as a subroutine in learning)

**Learning** tunes the parameters of the model.
Decomposing a Structure into Parts

• Why divide a structure into its pieces?
  – amenable to efficient inference
  – enable natural parameter sharing during learning
  – easier definition of fine-grained loss functions
  – clearer depiction of model’s uncertainty
  – easier specification of interactions between the parts
  – (may) lead to natural definition of a search problem

• A key step in formulating a task as a structured prediction
Scene Understanding

- **Variables:**
  - boundaries of image regions
  - tags of regions

- **Interactions:**
  - semantic plausibility of nearby tags
  - continuity of tags across visually similar regions (i.e. patches)

*(Li et al., 2009)*
Scene Understanding

• **Variables:**
  – boundaries of image regions
  – tags of regions

• **Interactions:**
  – semantic plausibility of nearby tags
  – continuity of tags across visually similar regions (i.e. patches)

(Li et al., 2009)
Word Alignment / Phrase Extraction

- **Variables (boolean):**
  - For each (Chinese phrase, English phrase) pair, are they linked?

- **Interactions:**
  - Word fertilities
  - Few “jumps” (discontinuities)
  - Syntactic reorderings
  - “ITG constraint” on alignment
  - Phrases are disjoint (?)

(Burkett & Klein, 2012)
Congressional Voting

- **Variables:**
  - Representative’s vote
  - Text of all speeches of a representative
  - Local contexts of references between two representatives

- **Interactions:**
  - Words used by representative and their vote
  - Pairs of representatives and their local context

(Stoyanov & Eisner, 2012)
Medical Diagnosis

- **Variables:**
  - content of text field
  - checkmark
  - dropdown menu

- **Interactions:**
  - groups of related symptoms (e.g. that are predictive of a disease)
  - social history (e.g. smoker) and symptoms
  - risk factors (e.g. infant) and lab results
Wikipedia Infobloxes

---

### Gryan Miers

**Personal information**

- **Date of birth**: 30 March 1999 (age 20)
- **Original team(s)**: Grovedale (GFL)
- **Draft**: No. 57, 2017 national draft
- **Debut**: Round 1, 2019, Geelong vs. Collingwood, at the MCG
- **Height**: 178 cm (5 ft 10 in)
- **Weight**: 78 kg (172 lb)
- **Position(s)**: Small forward

### Pather Panchali

**Directed by**: Satyajit Ray

**Screenplay by**: Satyajit Ray

**Based on**: Pather Panchali

by Bibhutibhushan Bandyopadhyay

**Starring**: Subir Banerjee

Kanu Banerjee

Chunibala Devi

Tuls Chakraborti

**Music by**: Ravi Shankar

**Cinematography**: Subrata Mitra

**Edited by**: Dulal Dutta

**Production company**: Government of West Bengal

**Distributed by**: Aurora Film Corporation (1955)

Edward Harrison (1958)

Merchant Ivory

**Release date**: 26 August 1955 (India)

### Changsha Kingdom

**Silk map unearthed from Mawangdui, showing Changsha and the neighboring kingdom of Nanyue.**

**Capital**: Linxiang (present-day Changsha)

**Government**: Monarchy

**History**:

- Established: 203/202 BC
- Extinction of the Wu family line: 157 BC
- Reestablishment under the Liu family: 155 BC
- Dissolution under Wang Mang: AD 9
- Restoration: AD 26
- Dissestablished: AD 33

### Space Invaders

**Developer(s)**: Taito

**Publisher(s)**:

- NA: Midway
- EU: Midway
- AU: Leisure & Allied Industries
- Atari, Inc. (home)

**Designers**: Tomohiro Nishikado

**Platform(s)**: Arcade, Atari 2600, Atari 5200, Atari 8-bit, MSX

**Release**: JP: June 1978

NA: July 1978

**Genre(s)**: Fixed shooter

**Mode(s)**: Single-player, 2 players alternating

**Cabinet**: Upright, cocktail

**Arcade system**: Taito 8080

**CPU**: 8080 @ 2 MHz

**Sound**: SN76477 @ 1.9958 MHz

**Display**: Mitsubishi MB14241, monochrome raster, vertical orientation, 224x256 resolution
Exercise: Wikipedia Infoboxes

Question:
Suppose you want to populate missing infobox fields.
1. What are the variables?
2. What are the interactions?

Answer:
ROADMAP
Roadmap by Contrasts

• **Model:**
  – locally normalized vs. globally normalized
  – generative vs. discriminative
  – treewidth: high vs. low
  – cyclic vs. acyclic graphical models
  – exponential family vs. neural
  – deep vs. shallow (when viewed as neural network)

• **Inference:**
  – exact vs. approximate (and which models admit which)
  – dynamic programming vs. sampling vs. optimization

• **Inference problems:**
  – MAP vs. marginal vs. partition function

• **Learning:**
  – fully-supervised vs. partially-supervised (latent variable models) vs. unsupervised
  – partially-supervised vs. semi-supervised (missing some variable values vs. missing labels for entire instances)
  – loss-aware vs. not
  – probabilistic vs. non-probabilistic
  – frequentist vs. Bayesian
Roadmap by Example

Whiteboard:

– Starting point: fully supervised HMM
– modifications to the model, inference, and learning
– corresponding technical terms of the result
SYLLABUS HIGHLIGHTS
Syllabus Highlights

The syllabus is located on the course webpage:

http://418.mlcourse.org
http://618.mlcourse.org

The **course policies** are **required** reading.
Syllabus Highlights

- **Grading 418**: 55% homework, 15% midterm, 25% final, 5% participation
- **Grading 618**: 50% homework, 15% midterm, 15% final, 5% participation, 15% project
- **Midterm Exam**: evening exam, Thu, Oct. 17
- **Final Exam**: final exam week, date TBD
- **Homework**: ~4 assignments
  - 6 grace days for homework assignments
  - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
  - No submissions accepted after 4 days w/o extension
  - Extension requests: see syllabus
- **Recitations**: Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings**: required, online PDFs, recommended for after lecture
- **Technologies**: Piazza (discussion), Autolab (programming), Canvas (quiz-style), Gradescope (open-ended)
- **Academic Integrity**:  
  - Collaboration encouraged, but must be documented
  - Solutions must always be written independently
  - No re-use of found code / past assignments
  - Severe penalties (i.e.. failure)
- **Office Hours**: posted on Google Calendar on “People” page
Lectures

• You should ask lots of questions
  – Interrupting (by raising a hand) to ask your question is strongly encouraged
  – Asking questions later (or in real time) on Piazza is also great

• When I ask a question…
  – I want you to answer
  – Even if you don’t answer, think it through as though I’m about to call on you

• Interaction improves learning (both in-class and at my office hours)
Textbooks

You are not required to read a textbook, but Koller & Friedman is a thorough reference text that includes a lot of the topics we cover.
Prerequisites

What they are:

1. Introductory machine learning. (i.e. 10-301, 10-315, 10-601, 10-701)

2. Significant experience programming in a general programming language.
   - Some homework may require you to use Python, so you will need to at least be proficient in the basics of Python.

3. College-level probability, calculus, linear algebra, and discrete mathematics.
Project (10-618 only)

• Goals:
  – Present an empirical comparison of competing methods for a task of your choice
  – For example:
    • compare models under the same inference technique
    • compare inference methods on the same model
    • compare learning methods on the same model
  – Deeper understanding of methods in real-world application

• Milestones: *(due in 2\textsuperscript{nd} half of semester)*
  1. Team Formation
  2. Proposal
  3. Midway Report
  4. Final Report
  5. Video Presentation
Q&A