Course Overview
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 Prediction

• The focus of most Intro ML courses is **classification**
  – Given observations: \( x = (x_1, x_2, ..., x_K) \)
  – Predict a (binary) **label**: \( y \)

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

• 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: \[ D = \{ x^{(n)}, y^{(n)} \}_{n=1}^{N} \]

<table>
<thead>
<tr>
<th>Sample 1:</th>
<th>Sample 2:</th>
<th>Sample 3:</th>
<th>Sample 4:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Sample 1 diagram" /></td>
<td><img src="image2" alt="Sample 2 diagram" /></td>
<td><img src="image3" alt="Sample 3 diagram" /></td>
<td><img src="image4" alt="Sample 4 diagram" /></td>
</tr>
</tbody>
</table>

Where:
- Each sample consists of a tuple \( x^{(n)}, y^{(n)} \)
- \( x^{(n)} \) represents the input sequence
- \( y^{(n)} \) represents the output (tagged sequence)

For example:
- Sample 1: \( x^{(1)} = (\text{time}, \text{flies}, \text{like}, \text{an}, \text{arrow}) \)
- Sample 2: \( x^{(2)} = (\text{time}, \text{flies}, \text{like}, \text{an}, \text{arrow}) \)
- Sample 3: \( x^{(3)} = (\text{flies}, \text{fly}, \text{with}, \text{their}, \text{wings}) \)
- Sample 4: \( x^{(4)} = (\text{with}, \text{time}, \text{you}, \text{will}, \text{see}) \)
Handwriting Recognition

Sample 1:

unexpected

Sample 2:

volcanic

Sample 2:

embraces

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

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

**Sample 1:**

1. \( y^{(1)} = \{ u, n, e, x, p, e, c, t, e, d \} \)
2. \( x^{(1)} = \{ \text{unexpected} \} \)

**Sample 2:**

1. \( y^{(2)} = \{ v, o, l, c, a, n, i, c \} \)
2. \( x^{(2)} = \{ \text{volcanic} \} \)

**Sample 3:**

1. \( y^{(3)} = \{ e, m, b, r, a, c, e, s \} \)
2. \( x^{(3)} = \{ \text{embraces} \} \)

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

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

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

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

- Pigeon: $x^{(1)}$, $y^{(1)}$
- Leopard: $x^{(3)}$, $y^{(3)}$
- Rhinoceros: $x^{(2)}$, $y^{(2)}$
- Llama: $x^{(4)}$, $y^{(4)}$
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

Classification
\[ \hat{y} = \arg\max_y p(y|x) \]
where \( y \in \{+1, -1\} \)

Structured Prediction
\[ \hat{y} = \arg\max_y p(y|x) \]
where \( y \in \mathcal{Y} \)
and \( |\mathcal{Y}| \) is very large
Structured Prediction

Data

Model

Objective

Inference

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.

**Learning** tunes the parameters of the model.

(Inference is usually called as a subroutine in learning.)
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)

Baseline. As an exact baseline, we compare against the results of Thomas et al. (2006). Their test-time Min-Cut algorithm is exact in this case: binary variables and a two-way classification.
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 Infoboxes
Exercise: Wikipedia Infoboxes

**Question:**
Suppose you want to populate missing infobox fields. What model would you pick for the job, and why?
A. Multiclass classifier
B. RNN
C. Graphical model

**Answer:**
A Visual Language for Variables and Interactions

GRAPHICAL MODELS
A joint distribution defines a probability $p(x)$ for each assignment of values $x$ to variables $X$. This gives the proportion of samples that will equal $x$. 

Sample 1: 
Sample 2: 
Sample 3: 
Sample 4: 
Sample 5: 
Sample 6: 

$X_0$ $X_1$ $X_2$ $X_3$ $X_4$ $X_5$ 
ψ₀ ψ₁ ψ₂ ψ₃ ψ₄ ψ₅ ψ₆ ψ₇ ψ₈ ψ₉ 
<START> time flies like an arrow
A joint distribution defines a probability $p(x)$ for each assignment of values $x$ to variables $X$. This gives the proportion of samples that will equal $x$. 

Sample 1:

Sample 2:

Sample 3:

Sample 4:
Sampling from a Joint Distribution

A **joint distribution** defines a probability \( p(x) \) for each assignment of values \( x \) to variables \( X \). This gives the **proportion** of samples that will equal \( x \).

<table>
<thead>
<tr>
<th>Sample 1:</th>
<th>Sample 2:</th>
<th>Sample 3:</th>
<th>Sample 4:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>time</strong></td>
<td><strong>time</strong></td>
<td><strong>flies</strong></td>
<td><strong>with</strong></td>
</tr>
<tr>
<td><strong>flies</strong></td>
<td><strong>flies</strong></td>
<td><strong>fly</strong></td>
<td><strong>time</strong></td>
</tr>
<tr>
<td><strong>like</strong></td>
<td><strong>like</strong></td>
<td><strong>with</strong></td>
<td><strong>you</strong></td>
</tr>
<tr>
<td><strong>an</strong></td>
<td><strong>an</strong></td>
<td><strong>their</strong></td>
<td><strong>will</strong></td>
</tr>
<tr>
<td><strong>arrow</strong></td>
<td><strong>arrow</strong></td>
<td><strong>wings</strong></td>
<td><strong>see</strong></td>
</tr>
</tbody>
</table>

The diagram illustrates the joint distribution with variables \( X_0, X_1, X_2, X_3, X_4, X_5 \) and words \( W_1, W_2, W_3, W_4, W_5 \). The chain structure indicates the order of sampling from the joint distribution.
Factors have local opinions ($\geq 0$)

Each black box looks at some of the tags $X_i$ and words $W_i$

Note: We chose to reuse the same factors at different positions in the sentence.
Factors have local opinions ($\geq 0$)

Each black box looks at some of the tags $X_i$ and words $W_i$

$$p(n, v, p, d, n, \text{time, flies, like, an, arrow}) = ?$$
Global probability = product of local opinions

Each black box looks at some of the tags $X_i$ and words $W_i$

$$p(n, v, p, d, n, \text{time, flies, like, an, arrow}) = \frac{1}{Z} (4 \times 8 \times 5 \times 3 \times \ldots)$$

Uh-oh! The probabilities of the various assignments sum up to $Z > 1$.
So divide them all by $Z$. 
Markov Random Field (MRF)

Joint distribution over tags $X_i$ and words $W_i$
The individual factors aren’t necessarily probabilities.

$$p(n, v, p, d, n, \text{time, flies, like, an, arrow}) = \frac{1}{Z}(4 \times 8 \times 5 \times 3 \times \ldots)$$
Hidden Markov Model

But sometimes we choose to make them probabilities. Constrain each row of a factor to sum to one. Now $Z = 1$.

$$p(n, v, p, d, n, \text{time, flies, like, an, arrow}) = \frac{1}{Z}(0.3 \times 0.8 \times 0.2 \times 0.5 \times \ldots)$$
Markov Random Field (MRF)

Joint distribution over tags $X_i$ and words $W_i$

$$p(n, v, p, d, n, \text{time, flies, like, an, arrow}) = \frac{1}{Z} (4 \times 8 \times 5 \times 3 \times \ldots)$$
Conditional Random Field (CRF)

Conditional distribution over tags $X_i$ given words $w_i$. The factors and $Z$ are now specific to the sentence $w$.

\[ p(n, v, p, d, n \mid \text{time, flies, like, an, arrow}) = \frac{1}{Z} (4 \times 8 \times 5 \times 3 \times \ldots) \]
Exercise: Wikipedia Infoboxes

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

**Answer:**

![Central Park Conservancy Infobox]

The Central Park Conservancy is a private, nonprofit park conservancy that manages Central Park under a contract with the City of New York and NYC Parks. The conservancy employs most maintenance and operations staff in the park. It effectively oversees the work of both the private and public employees under the authority of the publicly appointed Central Park administrator, who reports to the parks commissioner and the conservancy's president.[1] The Central Park Conservancy was founded in 1866 in the aftermath of Central Park's decline in the 1960s and 1970s.[2] Initially devoted to fundraising for projects to restore and improve the park, it took over the park's management duties in 1981.[3] The organization has invested more than $800 million toward the restoration and enhancement of Central Park since its founding.[4] With an endowment of over $200 million, consisting of contributions from residents, corporations, and foundations,[5] the Conservancy provides 75 percent of the Park's $565 million annual operating budget and is responsible for all basic care of the park.[6] The Conservancy also provides maintenance support and staff training programs for other public parks in New York City, and has assisted with the development of new parks, such as the High Line and Brooklyn Bridge Park.[7][8]
Exercise: Wikipedia Infoboxes

Populating Wikipedia Infoboxes

Q: Why do interactions appear between variables in this example?

A: Consider the test time setting:

– Author writes a new article (vector $x$)
– Infobox is empty
– ML system must populate all fields (vector $y$) at once
– Interactions that were seen (i.e. vector $y$) at training time are unobserved at test time – so we wish to model them
INFERENCE PROBLEMS
Inference

1. Data

\[ \mathcal{D} = \{ x^{(n)} \}_{n=1}^{N} \]

2. Model

\[ p(x | \theta) = \frac{1}{Z(\theta)} \prod_{C \in \mathcal{C}} \psi_C(x_C) \]

3. Objective

\[ \ell(\theta; \mathcal{D}) = \sum_{n=1}^{N} \log p(x^{(n)} | \theta) \]

4. Learning

\[ \theta^* = \arg\max_{\theta} \ell(\theta; \mathcal{D}) \]

5. Inference

1. Marginal Inference

\[ p(x_C) = \sum_{x': x_C = x_C} p(x' | \theta) \]

2. Partition Function

\[ Z(\theta) = \sum_{x} \prod_{C \in \mathcal{C}} \psi_C(x_C) \]

3. MAP Inference

\[ \hat{x} = \arg\max_{x} p(x | \theta) \]
5. Inference

1. Marginal Inference (#P-Hard)
   Compute marginals of variables and cliques
   \[ p(x_i) = \sum_{x' : x'_i = x_i} p(x' | \theta) \quad | \quad p(x_C) = \sum_{x' : x'_C = x_C} p(x' | \theta) \]

2. Partition Function (#P-Hard)
   Compute the normalization constant
   \[ Z(\theta) = \sum_{x} \prod_{C \in \mathcal{C}} \psi_C(x_C) \]

3. MAP Inference (NP-Hard)
   Compute variable assignment with highest probability
   \[ \hat{x} = \arg\max_x p(x | \theta) \]

4. Sampling (cf. convergence, variance)
   Draw a sample variable assignment
   \[ x \sim p(\cdot | \theta) \]
Q: But in deep learning we don’t need to solve these inference problems, right?

A: Wrong... it’s not that we don’t need to solve them, it’s that we can’t!

Questions you could ask your RNN-LM or seq2seq model:

1. What is the probability of the 7th token being ‘zebra’ (marginal inference)
2. For an unnormalized model, what is the normalization constant? (partition function)
3. What is the most probable output sequence? (MAP inference)
4. Give me 10 samples from the distribution.
SYLLABUS HIGHLIGHTS
Syllabus Highlights

The syllabus is located on the course webpage:
http://708.mlcourse.org  ...cs.cmu.edu...

The course policies are required reading.
Syllabus Highlights

• **Grading:** 45% homework, 35% project, 15% quizzes, 5% participation

• **Quizzes:** ~3 quizzes, during lecture/recitation time

• **Homework:** ~5 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),
  - Gradescope (homework),
  - Google Forms (polls),
  - Gather.Town (office hours),
  - Zoom (livestream),
  - Panopto (video recordings)

• **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, turning on your video, and waiting to be called on) to ask your question is strongly encouraged
  – Use the chat to ask questions in real time (TAs will be monitoring the chat and will either answer or interrupt the instructor)
  – Asking questions later 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)
# Homework

There will be 5 homework assignments during the semester. The assignments will consist of both conceptual and programming problems.

<table>
<thead>
<tr>
<th>HW1</th>
<th>Main Topic</th>
<th>Implementation</th>
<th>Application Area</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Properties of graphical models</td>
<td>NA</td>
<td>NA</td>
<td>written</td>
</tr>
<tr>
<td>HW2</td>
<td>Marginal inference and MLE</td>
<td>RNN + Tree CRF</td>
<td>natural language processing</td>
<td>written + programming</td>
</tr>
<tr>
<td>HW3</td>
<td>MAP inference and structured SVM</td>
<td>CNN + Ising model</td>
<td>computer vision</td>
<td>written + programming</td>
</tr>
<tr>
<td>HW4</td>
<td>MCMC</td>
<td>word embeddings + Gibbs sampler</td>
<td>topic modeling</td>
<td>written + programming</td>
</tr>
<tr>
<td>HW5</td>
<td>Variational Inference</td>
<td>variational inference</td>
<td>TBD</td>
<td>written + programming</td>
</tr>
</tbody>
</table>
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. 10-701, 10-715)
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.
Prerequisites

What they are:

1. Introductory machine learning. (i.e. 10-701, 10-715)
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.
Prerequisites

1. Introductory machine learning.
   (i.e. 10-701, 10-715)
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.
Prerequisites

What they are:

1. Introductory machine learning. (i.e. 10-701, 10-715)
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

• **Goals:**
  – Explore an interesting problem in your domain of interest
  – Employ graphical models for a structured prediction task or latent variable modeling
  – 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
  – Work in teams of 3 – 4 students

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