10-418 / 10-618 Machine Learning for Structured Data

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Course Introduction

Matt Gormley
Lecture 1
Aug. 26, 2019

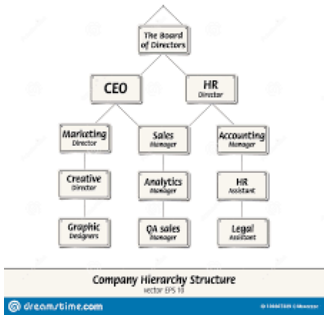
How to define a structured prediction problem

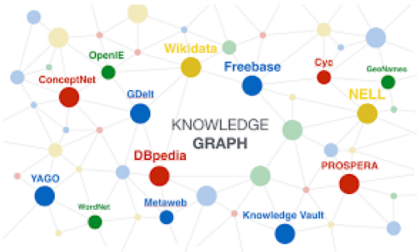
STRUCTURED PREDICTION

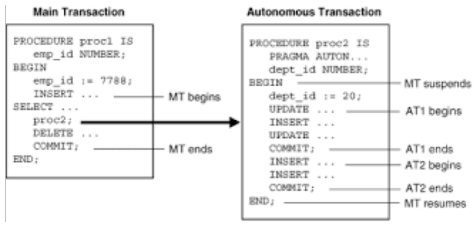
Structured vs. Unstructured Data

Structured Data Examples

- database entries
- transactional information
- wikipedia infobox
- knowledge graphs
- hierarchies



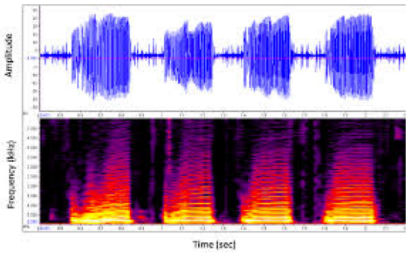




Unstructured Data Examples

- written text
- images
- videos
- spoken language
- music
- sensor data

مساء الخير! مرحبا بكم في الدرجة



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

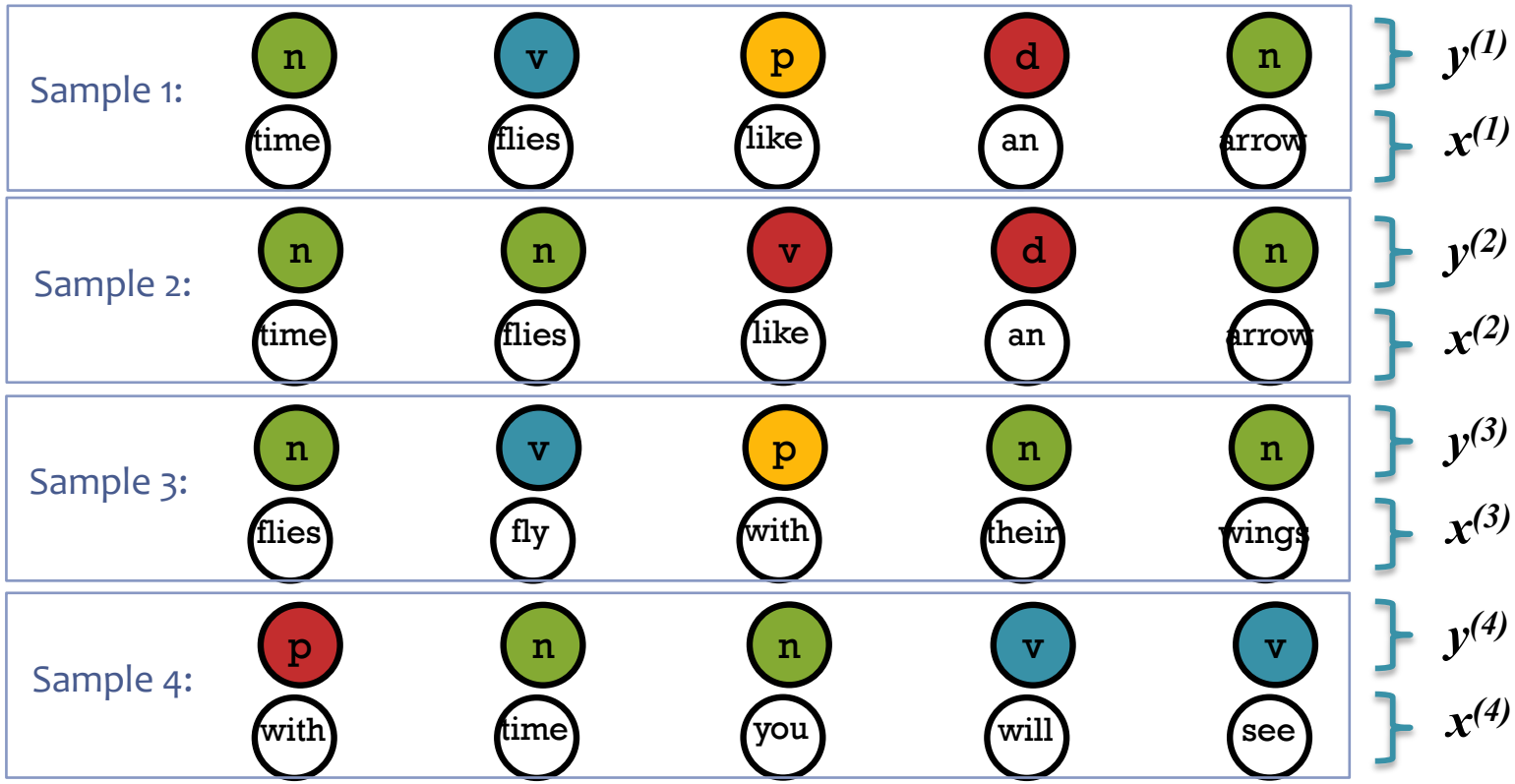
- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Object detection
  - Scene understanding
  - Machine translation
  - Protein sequencing

Part-of-Speech (POS) Tagging

Sample 1:	n	v	p	d	n
	time	flies	like	an	arrow
Sample 2:	n	n	v	d	n
	time	flies	like	an	arrow
Sample 3:	n	v	p	n	n
	flies	fly	with	their	wings
Sample 4:	p	n	n	v	v
	with	time	you	will	see

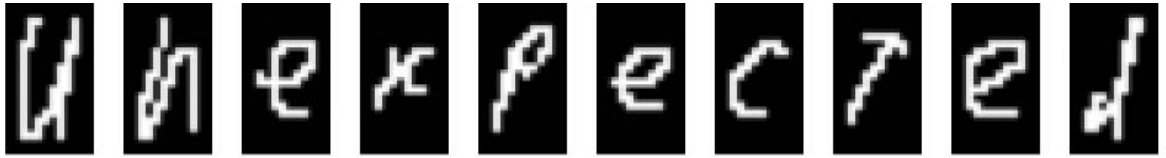
Dataset for Supervised Part-of-Speech (POS) Tagging

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

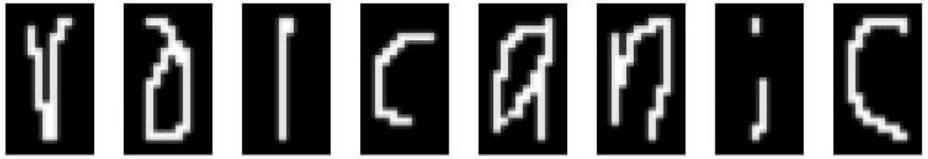


Handwriting Recognition

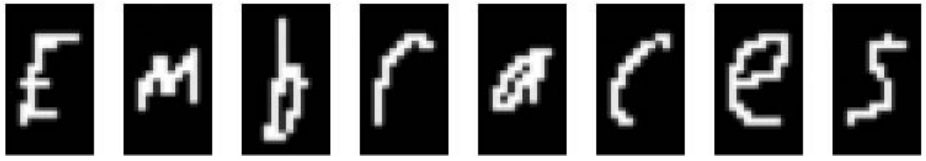
Sample 1: u n e x p e c t e d



Sample 2: v o l c a n i c



Sample 2: e m b r a c e s



Figures from (Chatzis & Demiris, 2013)

Dataset for Supervised Handwriting Recognition

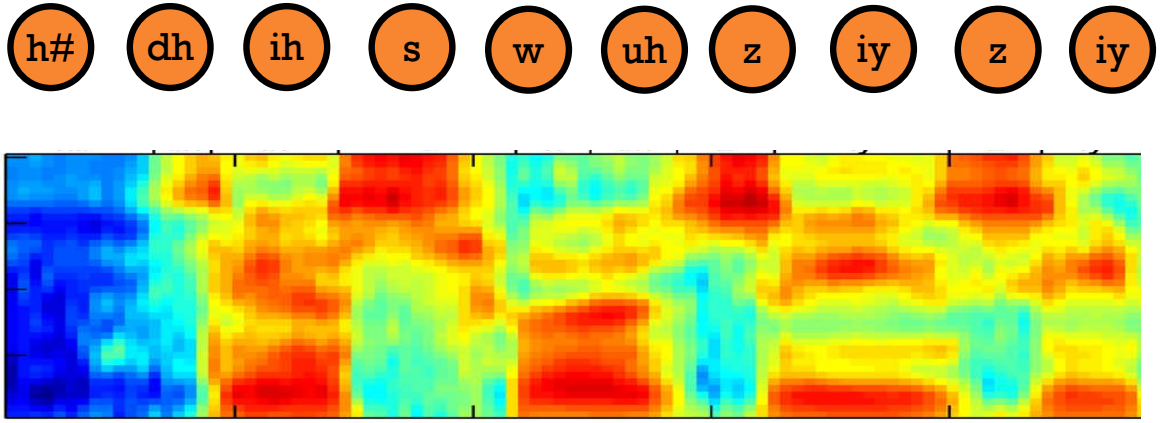
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Figures from (Chatzis & Demiris, 2013)

Dataset for Supervised Phoneme (Speech) Recognition

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

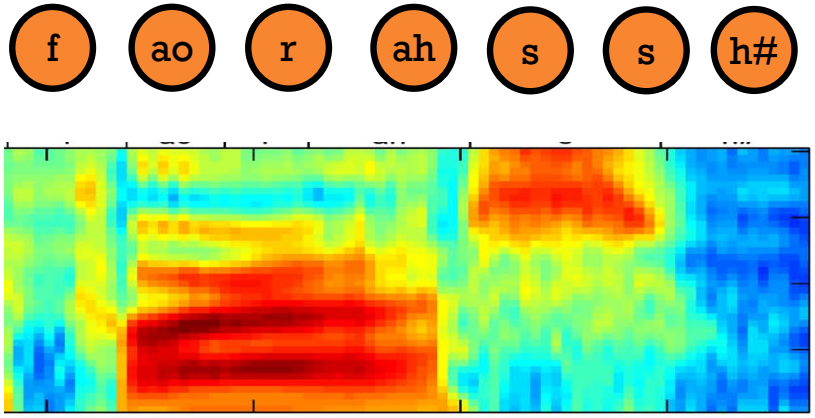
Sample 1:



$\boldsymbol{y}^{(1)}$

$\boldsymbol{x}^{(1)}$

Sample 2:



$\boldsymbol{y}^{(2)}$

$\boldsymbol{x}^{(2)}$

Figures from (Jansen & Niyogi, 2013)

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

$\boldsymbol{x^{(1)}}$

pigeon

$\boldsymbol{y^{(1)}}$

$\boldsymbol{x^{(2)}}$

rhinoceros

$\boldsymbol{y^{(2)}}$

$\boldsymbol{x^{(3)}}$

leopard

$\boldsymbol{y^{(3)}}$

$\boldsymbol{x^{(4)}}$

llama

$\boldsymbol{y^{(4)}}$

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

- Preprocess data into "patches"
- Posit a latent labeling $\boldsymbol{z}$ describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- $\boldsymbol{z}$ is not observed at train or test time

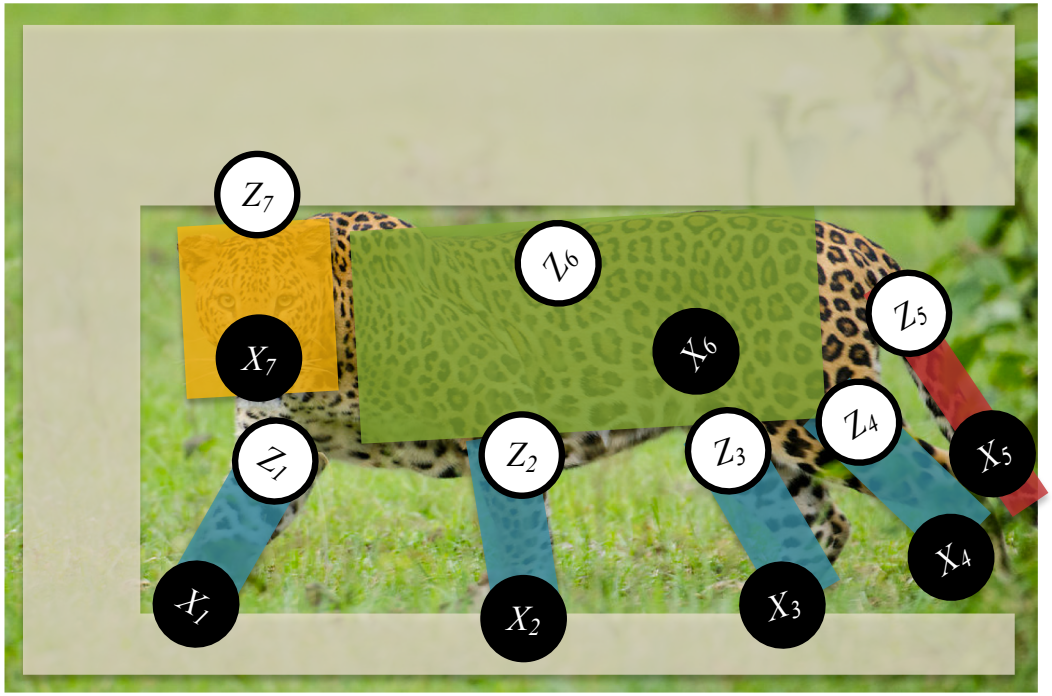
leopard

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

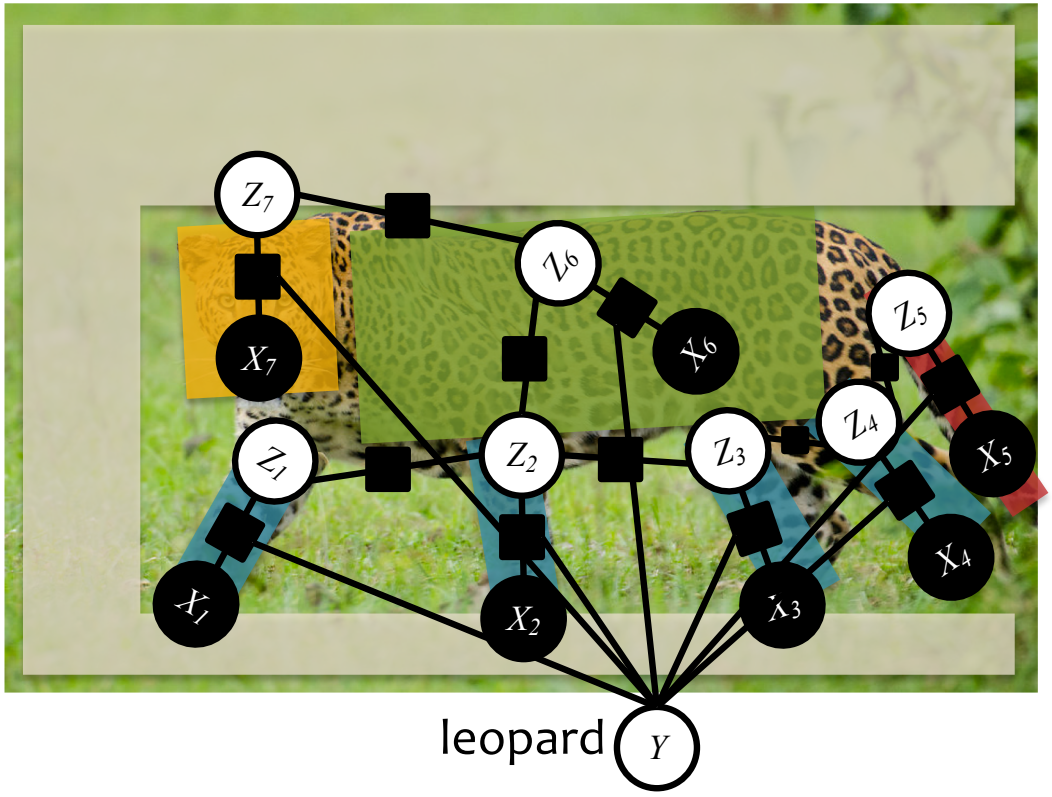


Case Study: Object Recognition

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

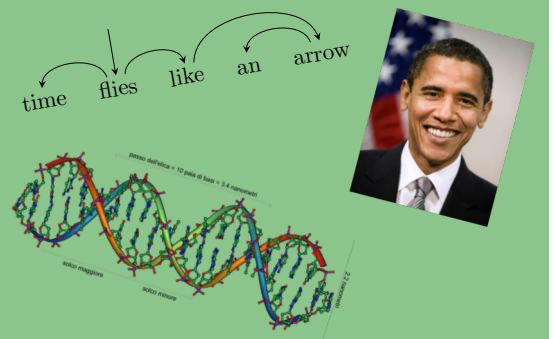
Preview of challenges to come…

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

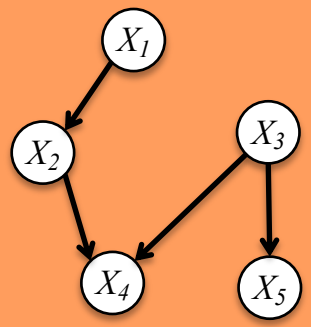
Classification	Structured Prediction
$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y\|\mathbf{x})$	$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, p(\mathbf{y}\|\mathbf{x})$
where $y \in \{+1, -1\}$	where $\mathbf{y} \in \mathcal{Y}$ and $\|\mathcal{Y}\|$ is very large

Structured Prediction

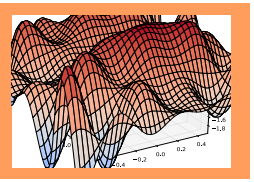
Data



Model



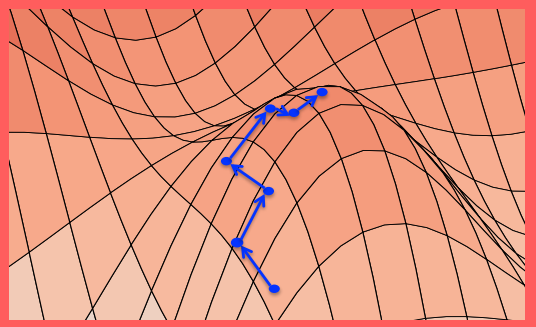
Objective



Inference

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

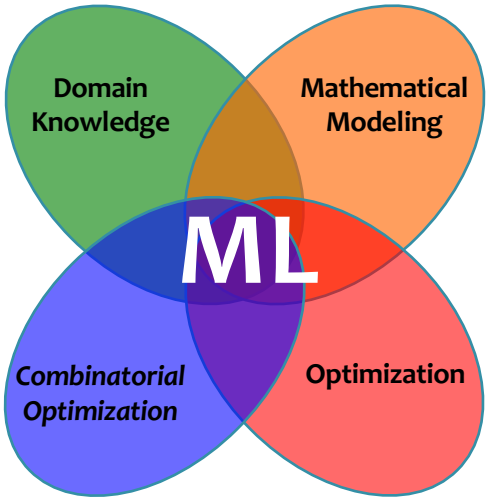
Learning



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)

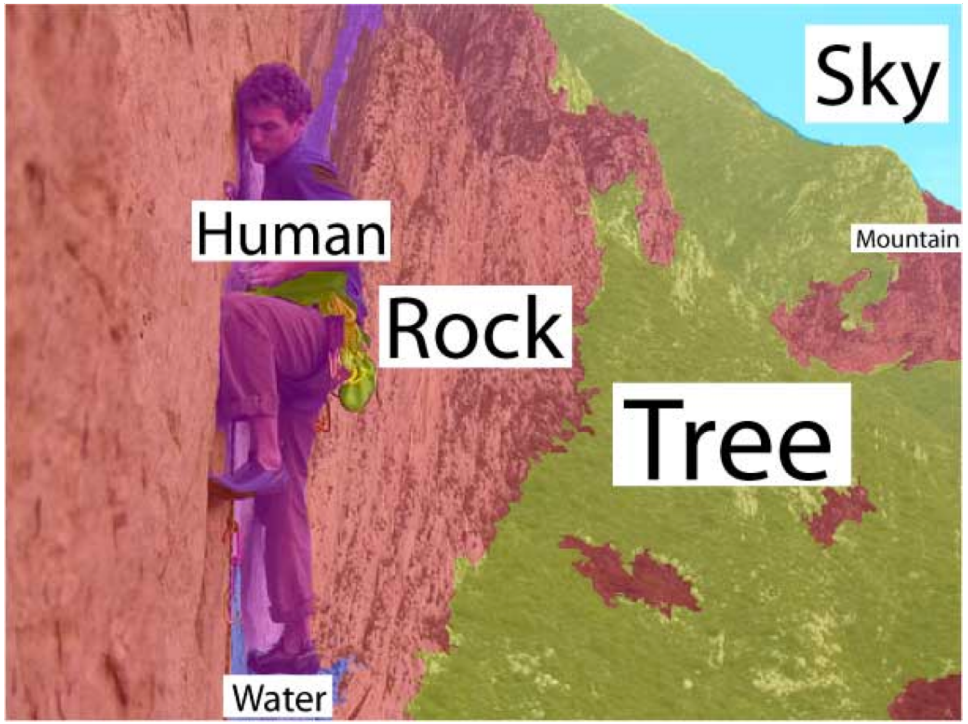
Decomposing a Structure into Parts

- 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)

Labels **with** top-down information

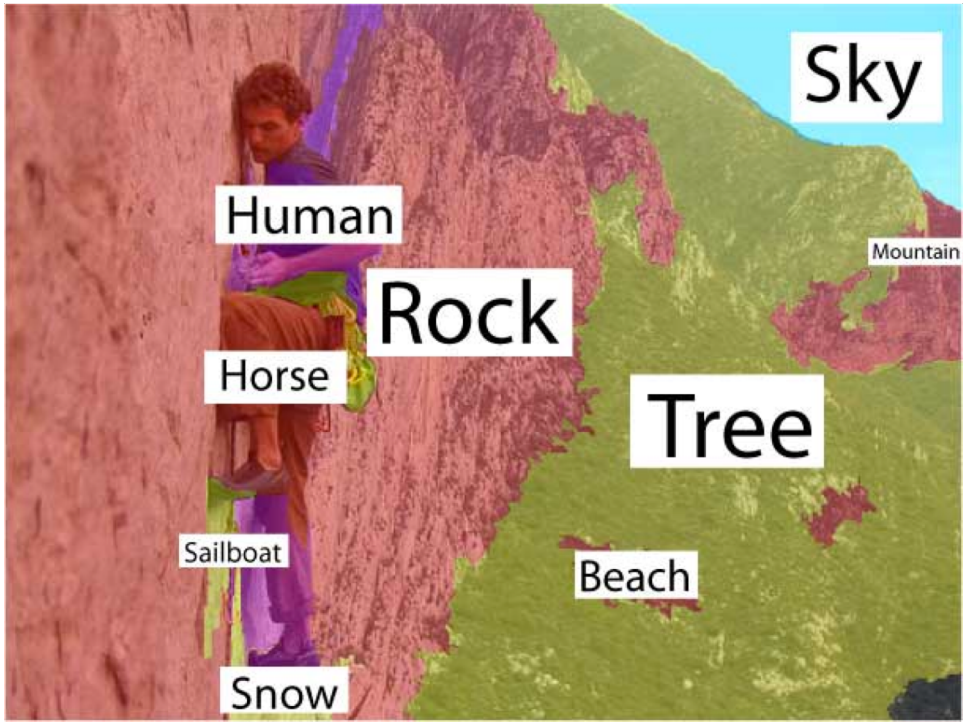


(Li et al., 2009)

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)

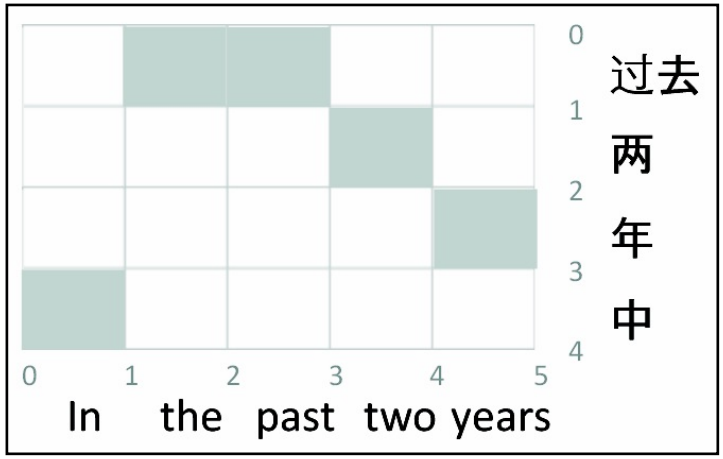
Labels **without** top-down information

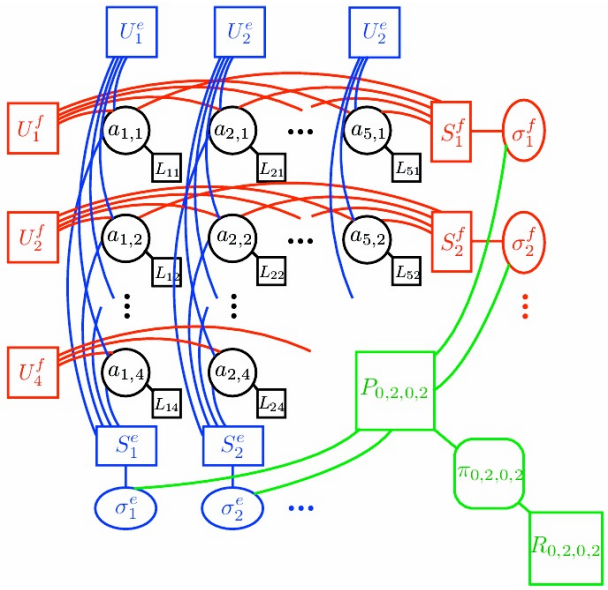


(Li et al., 2009)

Word Alignment / Phrase Extraction

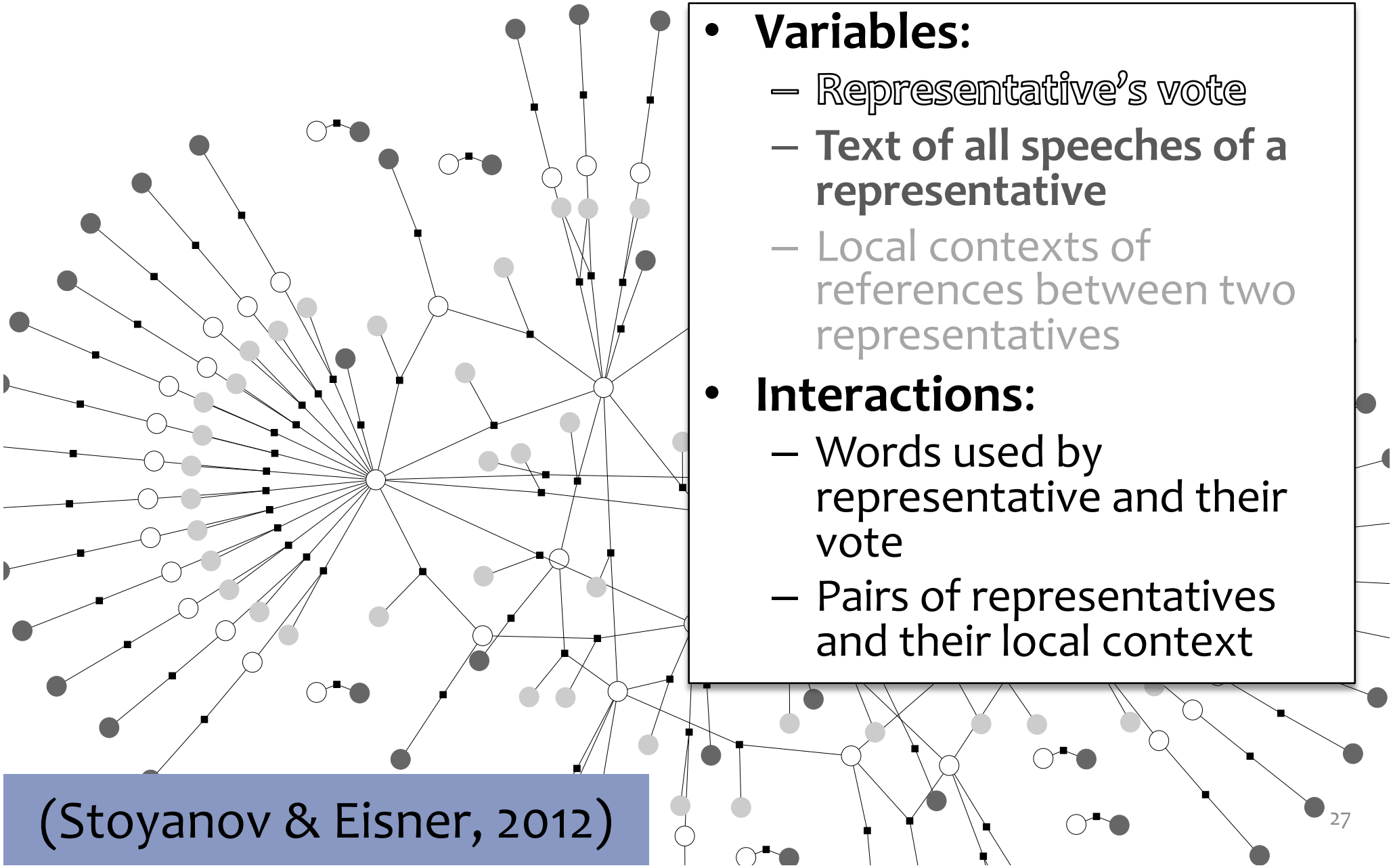
- **Variables (boolean):**
  - For each (Chinese phrase, English phrase) pair, are they linked?
- **Interactions:**
  - Word fertilities
  - Few "jumps" (discontinuities)
  - Syntactic reorderings
  - "ITG contraint" on alignment
  - Phrases are disjoint (?)





(Burkett & Klein, 2012)

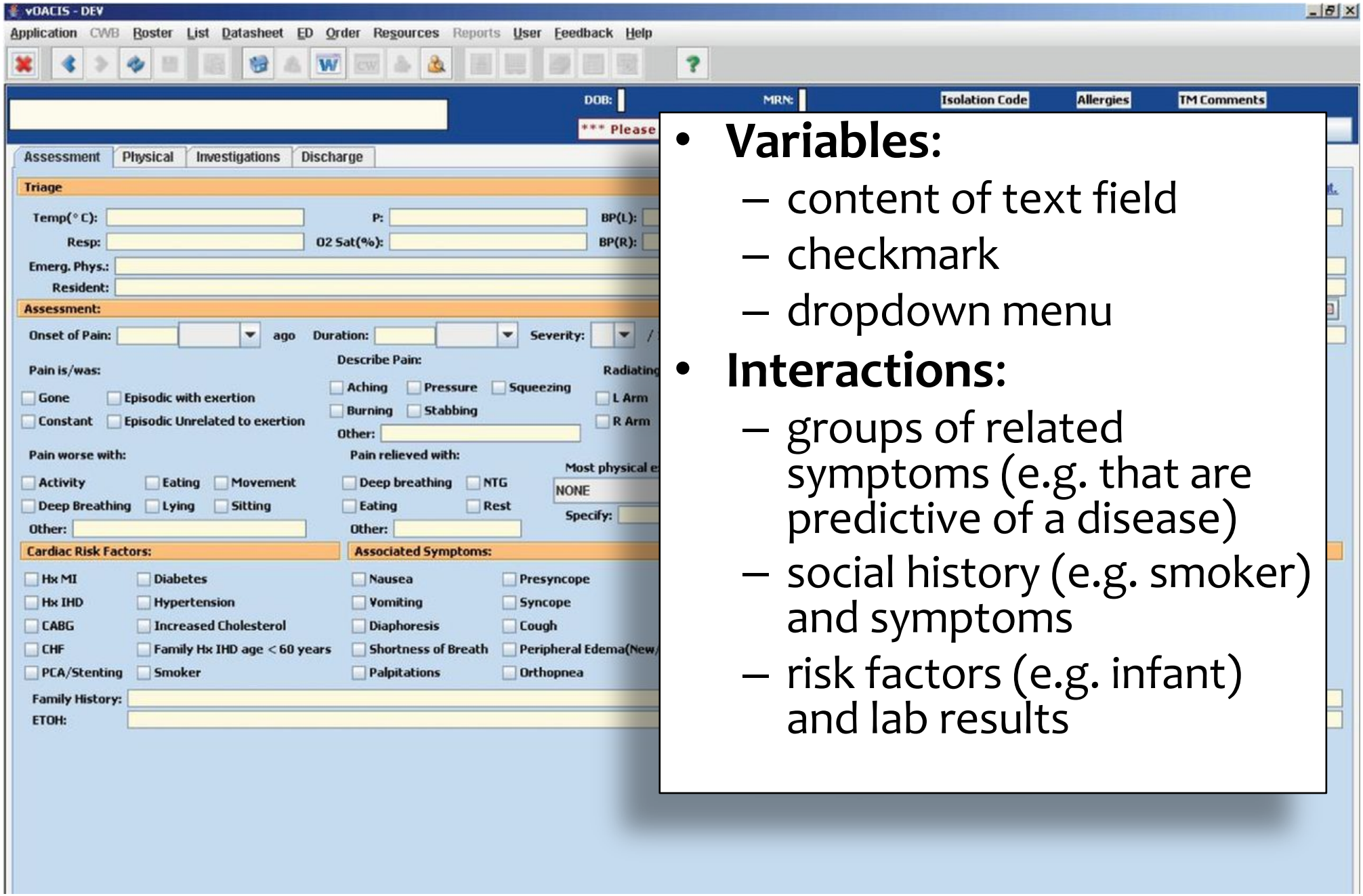
Congressional Voting



- **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 Infoboxes

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
Club information	
Current club	Geelong
Number	32

Playing career[1]

Years	Club	Games (Goals)
2019–	Geelong	19 (19)

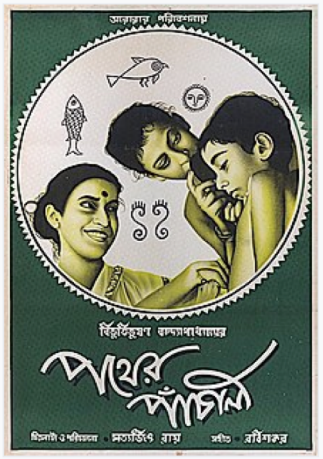
[1] Playing statistics correct to the end of round 17, 2019.

Career highlights

- AFL Rising Star nominee: 2019

Sources: AFL Tables, AustralianFootball.com

Pather Panchali



A poster of *Pather Panchali*

Directed by	Satyajit Ray
Screenplay by	Satyajit Ray
Based on	*Pather Panchali* by Bibhutibhushan Bandyopadhyay
Starring	Subir Banerjee Kanu Banerjee Karuna Banerjee Uma Dasgupta Chunibala Devi Tulsi Chakrabarti
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 Productions Sony Pictures Classics (1995)[a]
Release date	26 August 1955 (India)

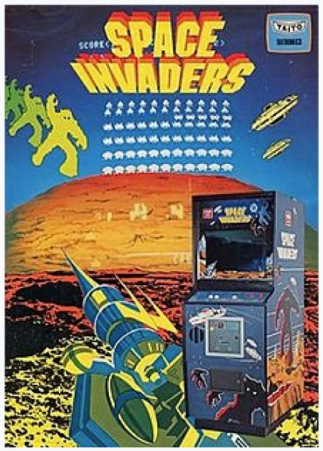
Changsha Kingdom
長沙國

203/202 BC–AD 33

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
• Disestablished	AD 33

Space Invaders



Promotional flyer

Developer(s)	Taito
Publisher(s)	JP: Taito NA: Midway EU: Midway[1] AU: Leisure & Allied Industries[2] Atari, Inc. (home)
Designer(s)	Tomohiro Nishikado
Platform(s)	Arcade, Atari 2600, Atari 5200, Atari 8-bit, MSX
Release	JP: June 1978[3] NA: July 1978
Genre(s)	Fixed shooter
Mode(s)	Single-player, 2 players alternating
Cabinet	Upright, cocktail[4]
Arcade system	Taito 8080[5]
CPU	8080 @ 2 MHz[5]
Sound	SN76477 @ 1.9968 MHz
Display	Fujitsu MB14241,[6] monochrome raster, vertical orientation, 224×256 resolution[5]

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

From Wikipedia, the free encyclopedia

Coordinates: 40.76424°N 73.97169°W

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 1980 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 1998.[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 $65 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]:45–46

Central Park Conservancy

Predecessor	Central Park Task Force, Central Park Community Fund
Founded	1980
Founders	Elizabeth Barlow Rogers, William Sperry Beinecke, Ed Koch, Gordon Davis, Andrew Stein (*ex officio*)
Tax ID no.	13-3022855
Location	New York City, U.S.
Coordinates	40.76424°N 73.97169°W
Area served	Central Park
Key people	Elizabeth W. Smith (President & CEO)
Website	centralparknyc.org

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

…cs.cmu.edu…

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

Syllabus Highlights

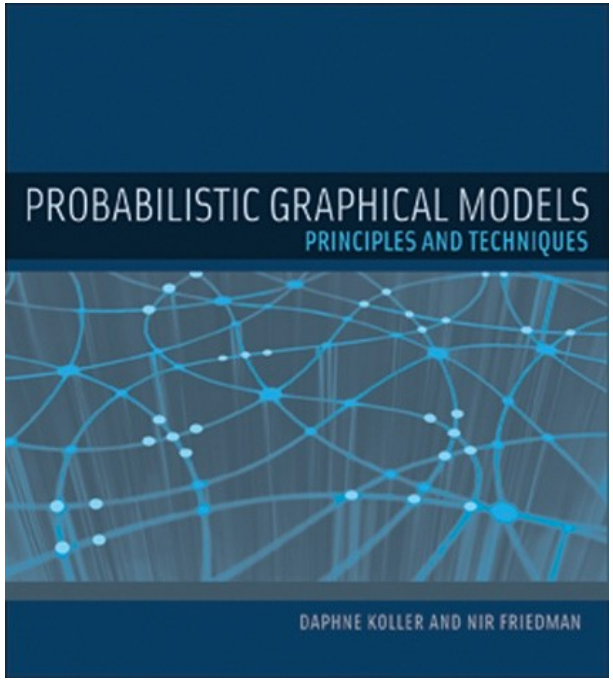
- **Grading 418**: 55% homework, 15% midterm, 25% final, 5% participation
- **Grading 618**: 50% homework, 15% midterm, 15% final, 5% participation, 15% project
- **Midterm Exam**: evening exam, Thu, Oct. 17
- **Final Exam**: final exam week, date TBD
- **Homework**: ~4 assignments
  - 6 grace days for 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 2nd half of semester*)
  1. Team Formation
  2. Proposal
  3. Midway Report
  4. Final Report
  5. Video Presentation

Q&A