COMPSCI 688: Probabilistic Graphical Models

Lecture 1: Course Overview

Dan Sheldon

Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Partially based on materials by Benjamin M. Marlin (marlin@cs.umass.edu) and Justin Domke (domke@cs.umass.edu)

Motivating Example

Motivating Example: Typo Corrector

Suppose we have a database of words with at most D letters, such as

$$\text{duck},\ \ \text{pile}, \text{an}\star\star,\ \ \text{dive},\ \ \cdots$$

where $\star$ is used to pad words with less than D letters (in this example $D = 4$).

Problem:

- We see “noisy” words: each letter has a 25% chance of corrupted to any random letter
- Given a noisy word, what is original clean work?

A probabilistic approach will have 3 steps...

Step 1: Distribution of Words

- Build a distribution $p(x)$ over all length D sequences in the database
- Each sequence represented by $x = (x_1, x_2, \cdots, x_D)$ with $x_i \in \{\mathrm{a}, \mathrm{b}, \cdots, \mathrm{z}, \star\}$.
- $p(x)$ is a measure of how likely x is to occur as an English word.

Example

$$\begin{aligned} p(\mathrm{a}, \mathrm{a}, \mathrm{a}, \mathrm{a}) &= 0.000001 \\ p(\mathrm{a}, \mathrm{a}, \mathrm{a}, \mathrm{b}) &= 0.000002 \\ &\vdots \\ p(\mathrm{t}, \mathrm{a}, \mathrm{c}, \mathrm{o}) &= 0.00531 \\ &\vdots \end{aligned}$$

Step 2: Conditional Distribution of Noisy Words

We build a conditional distribution $p(y|x)$ of the "noisy" sequences y given "clean" ones x. In this case, the conditional distribution is

$$p(y|x) = \prod_{i=1}^{D} \left(0.75 \times \mathbb{I}[y_i = x_i] + 0.25 \times \frac{1}{27} \right).$$

- $\mathbb{I}[\cdot]$ is indicator
- Each position i corrupted independently:
  - with probability 0.75, keep x_i
  - with probability 0.25, select a random letter

Step 3: Combine to Make Predictions

Given noisy sequence y, want to predict a clean sequence x. Bayes' rule says:

$$p(x|y) = \frac{p(x)p(y|x)}{p(y)}.$$

Predict the most likely x as

$$\arg\max_x p(x|y) = \arg\max_x p(x)p(y|x),$$

E.g., use brute force to search over all x. But wait:

- how much time?
- how big does our data set need to be?

Brute Force Algorithm

1. For all x, compute $\mathrm{score}(x) = p(x)p(y|x)$.
2. Return x with highest score.

- How much time?
- Is there a smarter algorithm?
- How many free parameters in $p(x)$?
- How big would our data set need to be to estimate it?

Lesson: for large D, we need structure

27^D is big

D	$(27)^D$
1	27
2	729
5	$14,348,489$
10	$205,891,132,094,649$
...	...

Graphical Models = Factorized Distributions

Suppose $p(x)$ has a factorized form:

$$p(x) = f(x_1, x_2) f(x_2, x_3) \cdots f(x_{D-1}, x_D).$$

What would this buy us?

- Statistics: only $\approx (D-1)(27)^2$ free parameters
- Computation: can find the MAP solution in $\approx (D-1)(27)^2$ operations (dynamic programming)

Factorization is great! But when is it “valid”? When $p(x)$ has conditional independencies (CIs)...

Course Overview

Probabilistic Graphical Models

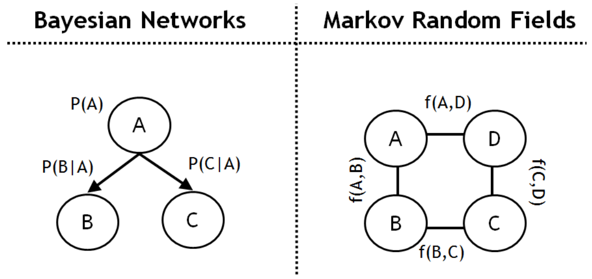
- Main AI tool for representing, constructing and reasoning about large-scale, high dimensional systems under uncertainty.
- Model **large-scale, complex** systems as collection of **simpler, locally interacting** subsystems
- Apply probabilistic modeling (cf. typo corrector) to a range of problems **in high dimensions**
- **Many applications:** speech recognition, image recognition and labeling, image modeling, action recognition, modeling sensor networks, social network analysis, recommender systems, evolutionary biology, proteomics and genomics, medical decision making, information extraction, text modeling, and many more...

Main idea

- **Represent** the structure of high-dimensional joint probability distributions using graphs (graph structure models conditional indpendencies)
- **Learn** the distribution from data
- Perform **inference** to efficiently answer probability queries (i.e., compute conditional distributions) using the graph structure

Bayesian Networks and Markov Networks

The two most common types of probabilistic graphical models are Bayesian Networks and Markov Networks.



Probabilistic Graphical Models vs Machine Learning

- Probabilistic graphical models (PGMs) are a sub-topic of machine learning
- PGM models exist for essentially all main ML tasks: classification, regression, clustering, dimensionality reduction, ...
- Many classical ML models are special cases of PGMs
- Knowledge of PGMs allows you to build customized models for dealing with complex, uncertain, and partially observed data.

Course Goals

The aim of this course is to develop the knowledge and skills necessary to effectively design, implement and apply probabilistic graphical models to solve real-world problems. The course will cover:

- Bayesian and Markov networks
- Exact and approximate inference methods for answering probability queries and making predictions
- Estimation of the parameters and structure of graphical models from data

Prerequisites

Formally, none. However, we will move quickly through a lot of material. Familiarity with the following material is highly recommended:

- Probability and statistics
- Calculus and linear algebra
- Basic algorithms and data structures
- Numerical optimization
- Machine Learning

Programming and Computing

- Need access to computing to complete regular assignments (any moderately recent laptop/desktop should do).
- Python **strongly recommended**. I recommend using an Anaconda distribution.

Logistics

Logistics and course details:

- Instructor: Dan Sheldon
- TAs: Jinlin Lai, Shakir Sahibul
- Lectures: Tu/Th 4:00-5:15pm
- Instructor Office Hours: TBD
- Course Website: https://people.cs.umass.edu/~sheldon/teaching/cs688/index.html
- Discussion: Piazza
- Homework Submission: Gradescope
- Course e-Mail: Piazza private message

Textbooks

There is no required book, but optional supplementary readings will be assigned in two books:

- MLPP: *Machine Learning: A probabilistic Perspective*. Murphy. (Primary; free eBook for UMass students)
- PGM: *Probabilistic Graphical Models* by Koller and Friedman. (Supplemental)

The readings will cover similar material.

Evaluation

The evaluation for the course will be based on quizzes, assignments, and a final exam.

- Homework Assignments 60%
- Final Exam 30%
- Quizzes 10%

Course Policies

This is a large class. Course policies are applied with exceptions only in exceptional situations. Read the course syllabus for details of:

- Homework submission and late days
- Homework collaboration
- Academic honesty
- Regrading

Echo360 Lecture Capture

- Lectures are recorded and will be available after 3–4 days
- There is form to request access sooner (see course webpage)
- Usually 1–2 recordings per semester fail

Probability Review

Discrete Probability Distribution

- A discrete distributions models a random experiment (e.g., a coin flip, roll of the die, shot of an arrow) with a finite or countable number of outcomes
- The *sample space* Ω is the set of outcomes
- A probability distribution P on Ω assigns a non-negative real number or *atomic probability* $p(\omega)$ to each outcome $\omega \in \Omega$, such that

$$p(\omega) \geq 0, \quad \sum_{\omega \in \Omega} p(\omega) = 1$$

Events

- An **event** $A \subseteq \Omega$ is a subset of the sample space
- The **probability** of an event if the sum of the probabilities of its outcomes:

$$P(A) = \sum_{\omega \in A} p(\omega)$$

- Note: events are the *only things* that have probabilities, ever
- When Ω is not discrete, we need to be more careful about defining events and their probabilities (measure theory)

Example

Imagine the random experiment of rolling a fair six-sided die:

- Sample Space: $\Omega = \{1, 2, 3, 4, 5, 6\}$
- Consider the events $A = \{1, 2\}$ and $B = \{2\}$.
- Then $P(A) = 1/3$, $P(B) = 1/6$
- Also, $P(A \cap B) = 1/6$, $P(A \cup B) = 1/3$

Random Variables

Events can be cumbersome. With PGMs, we'll usually work with random variables.

A random variable X is a mapping $X : \Omega \to D$

- D is some set (e.g., the integers)
- Notation: $D = \mathrm{Val}(X)$, the set of values of X

A random variable partitions Ω:

- For each $x \in D$, we have the event $[X = x] = \{\omega : X(\omega) = x\}$
- It's probability is

$$P(X = x) = P(\{\omega : X(\omega) = x\}) = \sum_{\omega : X(\omega) = x} p(\omega)$$

Example: Rolling two Six-Sided Dice

Say X is the sum of the two fair dice. Then

- Sample Space: $\Omega = \{(\omega_1, \omega_2) | \omega_1, \omega_2 \in \{1, ..., 6\}\} = \{(1,1), (1,2), ..., (6,6)\}$
- Domain $D = \mathrm{Val}(X) = \{2, ..., 12\}$
- Mapping: $X((\omega_1, \omega_2)) = \omega_1 + \omega_2$
- Example Event: $\{X = 4\} = \{(1,3), (2,2), (3,1)\}$

Probability Mass Function

The *probability mass function* (PMF) of a discrete random variable X is a function p_X that gives the probability of the event $[X = x]$ for every $x \in \mathrm{Val}(X)$:

$$p_X(x) = P(X = x)$$

Thought experiment: the PMF also satisfies the defintion of a discrete probability distribution

$$p_X(x) \geq 0, \qquad \sum_{x \in \mathrm{Val}(X)} p(x) = 1$$

Why didn't we just use $\mathrm{Val}(X)$ as the sample space?

Next Time

- A bit more probability
  - joint distributions
  - rules of probability
  - independence and conditional independence
- Bayes' nets