CSC 412/2506: Probabilistic Machine Learning

Week 1: Introduction and Preliminaries

Ohad Shamir

University of Toronto

First part of the lecture

- What is this class about?
- Administration details

This course

- Introduction to **probabilistic** machine learning (PML).
- We introduce many fundamental concepts of PML.
- Aimed at advanced undergrad and master level graduate students.
- We will use **a lot of** multivariate calculus, probability, and linear algebra.

Course Information

Course Website: `https://ohadshamir2.github.io/csc412_f26/`

Main source of information is the course webpage; check regularly!

We will also use Quercus for **announcements & grades etc**.

We will use Piazza for **discussions**.

- Sign up via quercus or: `https://piazza.com/utoronto.ca/fall2026/csc412`
- Your grade **does not depend on your participation on Piazza**. It's just a good way for asking questions, discussing with your instructor, TAs and your peers. We will only allow questions that are related to the course materials/assignments/exams.

Course Information

- This course has one section: Lecture Tuesday 9-11pm (MP 134), Tutorial Thursdsay 10-11 (ES B149)
  - NOTE: Substantially same course also offered in winter semester by Prof. Erdogdu
- Instructor office hours are on Tuesday 1-3pm, Pratt 290G.
- TA office hours will be announced with each assignment.
- Questions during lectures/tutorials are always welcome!

Course Information

- While cell phones and other electronics are not prohibited in lecture, please do not use them in a manner which interferes with your fellow students.
- Please do not distribute course materials that are not publicly available!
- Lecture slides and notes will be posted on the course webpage. Please do let us know about typos you notice and/or any suggestions you might have.

Course Information

- The absence declaration will be considered sufficient to indicate absence. It is student's responsibility to inform the instructor in a timely manner if extensions to assignments are required.
- For accessibility services: If you require additional academic accommodations, please contact UofT Accessibility Services as soon as possible, `studentlife.utoronto.ca/as`.

Literature

Recommended (but not necessary) readings will be given for each lecture. The following will be useful throughout the course:

- Murphy: “Machine Learning: A Probabilistic Perspective”, 2012.
- Murphy: “Probabilistic Machine Learning: An introduction”, 2022.
- Murphy: “Probabilistic Machine Learning: Advanced topics”, 2023.
- Bishop: “Pattern Recognition and Machine Learning”, 2006.
- Hastie, Tibshirani, and Friedman: “The Elements of Statistical Learning”, 2009.

There are lots of freely available, high-quality ML resources.

Requirements and Marking

- Three homework assignments
  - Combination of pen & paper derivations and coding exercises
  - Equally weighted for a total of 30%
- Midterm
  - 3 November, in class
  - ~ 2 hours
  - Worth 30% of course grade
- Final Exam
  - ~ 3 hours
  - Date and time TBA
  - Worth 40% of course grade
- Exam questions are conceptual/theoretical; no coding.
- **Everybody must take the final exam! No exceptions.**

More on Assignments

- Collaboration on the assignments is allowed. After attempting the problems on an individual basis, you may discuss and work together on the homework assignments with up to two classmates. However, you must write your own code and write up your own solutions individually and explicitly name any collaborators at the top of the homework.
- No strict policy against using LLMs in assignments.

More on Assignments

- Collaboration on the assignments is allowed. After attempting the problems on an individual basis, you may discuss and work together on the homework assignments with up to two classmates. However, you must write your own code and write up your own solutions individually and explicitly name any collaborators at the top of the homework.
- No strict policy against using LLMs in assignments.
- The schedule of assignments are posted on the course webpage.
- Assignments should be handed in (via Crowdmark) by deadline; a late penalty of 10% **per day** will be assessed thereafter (up to 3 days, then submission is blocked).
- Extensions will be granted only in special situations, and you will need to fill out absence declaration form and **inform the instructor** or have documentation from the accessibility services.
- You will be using Python on assignments.

Related Courses

- STA314 and CSC311: Intro ML (we build on these courses)
- STA414/2104: Mostly same material
- **CSC412/2506: This course**
- CSC413: Neural networks and deep learning
- STA302: Linear regression and classical statistics
- CSC2515: Advanced CSC311
- Various topics and seminar style courses offered at DoSS and DCS
  - CSC2541: (New) topics course on learning theory

Provisional Calendar (tentative)

- week 1, Sep 8/10:
  - Introduction
  - Probabilistic models (exponential families, MLE)
- week 2, Sep 15/17:
  - Statistical Decision Theory
  - Directed graphical models I (DAGs)
- week 3, Sep 22/ 24:
  - Directed graphical models II (DAGs)
  - Markov Random Fields
  - Assignment 1 release on Sep 22
- week 4, Sep 29/ Oct 1:
  - Exact inference
  - Message passing

Provisional Calendar (cont'ed)

- week 5, Oct 6/8:
  - Sampling
  - Assignment 1 due on Oct 5
  - Assignment 2 release on Oct 6
- week 6, Oct 13/15:
  - Hidden Markov models
  - Variational inference I
- week 7, Oct 20/22:
  - Variational inference II
  - Variational Auto-encoders
  - Assignment 2 due on Oct 19
- week 8:
  - Reading week – no classes
- week 9, Nov 3:
  - Midterm exam on Nov 3
  - Assignment 3 release on Nov 3

Provisional Calendar (cont'ed)

- week 10, Nov 10/12:
  - Probabilistic PCA
  - EM algorithm
- week 11, Nov 17:
  - Bayesian regression
  - Kernel methods
- week 12, Nov 24/26:
  - Gaussian processes
  - Diffusion models I
  - Assignment 3 due on Nov 23
- week 13, Dec 01 :
  - Diffusion models II
  - Final exam review
- TBD: Final Exam

What is machine learning?

What is machine learning?

- Often, it is difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech
  - system needs to adapt to a changing environment (e.g. spam detection)
- Machine learning approach: program an algorithm to automatically learn from data, or from experience.

What is machine learning?

- It is similar to statistics...
  - Both try to uncover patterns in data.
  - Both share many of the same core algorithms and models.
  - Both draw heavily on calculus, probability, and linear algebra.

What is machine learning?

- It is similar to statistics...
  - Both try to uncover patterns in data.
  - Both share many of the same core algorithms and models.
  - Both draw heavily on calculus, probability, and linear algebra.
- But machine learning is not statistics!
  - Statistics is more concerned with helping scientists and policymakers draw good conclusions; ML is more concerned with building systems that have a desired behavior.
  - Statistics puts more emphasis on interpretability and mathematical rigour; ML puts more emphasis on predictive performance, scalability, and autonomy.

What is machine learning?

- Types of machine learning
  - **Supervised learning:** Given input-output pairs $(x^{(i)}, y^{(i)})$, the goal is to learn the mapping f from inputs x to outputs y.
  - **Unsupervised learning:** Given unlabeled data instances $x^{(i)}$, the goal is to find relations among inputs, which can be used for making predictions, decisions. The objective can vary.
  - **Semi-supervised learning:** We are given only a limited amount of labeled data, i.e. $(x^{(i)}, y^{(i)})$ pairs, but lots of unlabeled $x^{(i)}$'s.
  - **Reinforcement learning:** learning system receives a reward signal, tries to learn to maximize the reward signal.

What is machine learning?

- Types of machine learning
  - **Supervised learning:** Given input-output pairs $(x^{(i)}, y^{(i)})$, the goal is to learn the mapping f from inputs x to outputs y.
  - **Unsupervised learning:** Given unlabeled data instances $x^{(i)}$, the goal is to find relations among inputs, which can be used for making predictions, decisions. The objective can vary.
  - **Semi-supervised learning:** We are given only a limited amount of labeled data, i.e. $(x^{(i)}, y^{(i)})$ pairs, but lots of unlabeled $x^{(i)}$'s.
  - **Reinforcement learning:** learning system receives a reward signal, tries to learn to maximize the reward signal.

Note that these can all be modeled as estimating distributions from data: $p(y|x)$, $p(x)$, $p(x, y)$! This is the main focus of this course.

History of machine learning

- 1957 — Perceptron algorithm (implemented as a circuit!)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him
- 1969 — Minsky and Papert's book *Perceptrons* (limitations of linear models)
- 1970s — "AI Winter", a time of pessimism and low funding
- 1980s — Some foundational ideas
  - Connectionist psychologists explored neural models of cognition
  - 1984 — Leslie Valiant formalized the problem of learning as PAC learning
  - 1988 — Backpropagation (re-)discovered by Geoffrey Hinton and colleagues
  - 1988 — Judea Pearl's book *Probabilistic Reasoning in Intelligent Systems* introduced Bayesian networks

History of machine learning

- Late 1980's — a shorter "AI Winter"
- 1990s – start of a golden age for ML research
  - Markov chain Monte Carlo
  - variational inference
  - kernels and support vector machines
  - boosting
  - convolutional networks
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s and beyond — deep learning
  - 2012 — neural nets smashed previous records in speech-to-text and object recognition
  - increasing adoption by the tech industry
  - 2016 — AlphaGo defeated the human Go champion
  - 2020 – Alphafold for protein structure prediction
  - 2023 – Chatgpt

Computer vision: Object detection, semantic segmentation, pose estimation, and almost every other task is done with ML.

Figure 4. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

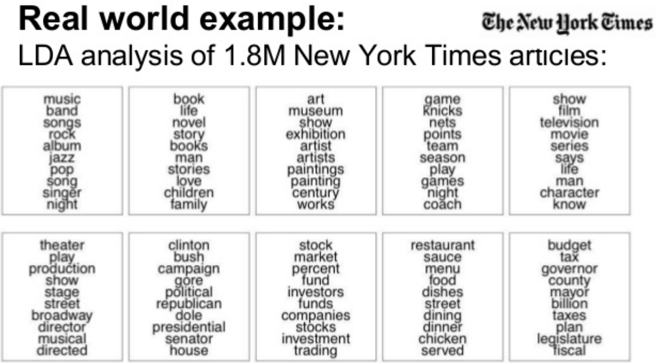
Instance segmentation - Link

DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

COCOQA 5078
How many leftover donuts is the red bicycle holding?
Ground truth: three
IMG+BOW: two (0.51)
2-VIS+BLSTM: three (0.27)
BOW: one (0.29)

Speech: Speech to text, personal assistants, speaker identification...

NLP: Translation, sentiment analysis, topic modeling, spam filtering.



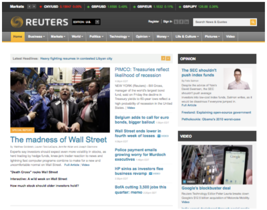
Finding Structure in Data

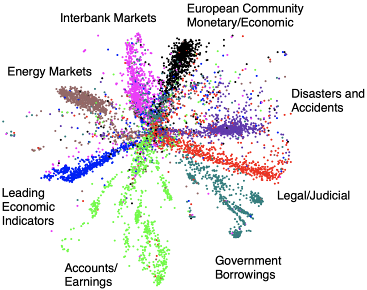
Take a large newswire corpus, for example. A simple model based on the word counts of webpages

$$P(x) = \frac{1}{Z} \sum_h \exp[x^\top W h]$$

could learn to discretize data into topics. In this case, our topics are our **hidden** (or **latent**) variables.

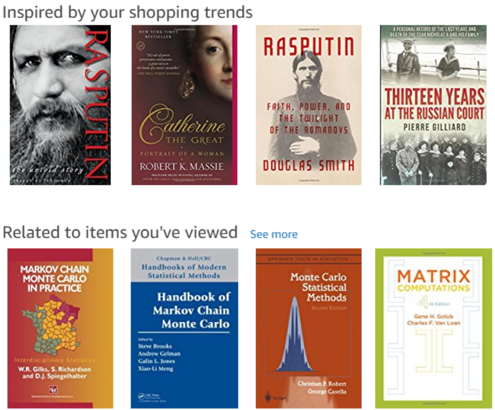
Vector of word counts on a webpage





804,414 newswire stories

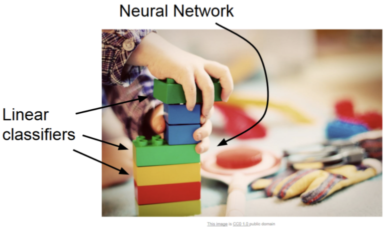
E-commerce & Recommender Systems : Amazon, Netflix, . . .

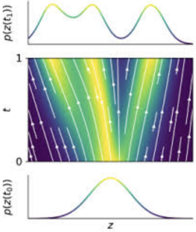


Why this class?

- This class complements CSC 311.
- We discuss fundamental probabilistic ideas in ML.
- Probabilistic latent variable models and decision theory can cover a wide range of machine learning models.
- **This is not a deep learning course!** But the principles you will learn are essential to understand deep learning models.

What can you build with these tools?





- Naive Bayes, Mixture of Gaussians, Logistic Regression, Bayesian Linear Regression, Hidden Markov Models, Factor Analysis
- Neural network classifiers, LSTMs, RNNs, Transformers, Convnets, Neural ODEs
- Variational Autoencoders, Generative Adversarial Networks, Normalizing Flows, Diffusion Models
- ...

Implementing machine learning systems

There are many neural net frameworks, e.g. PyTorch, TensorFlow. Why it is useful to study the **fundamentals** of machine learning?

The fundamentals gives you:

- a better understanding of how different algorithms and models work, and how to choose the appropriate ones for a given task.
- a deeper understanding of the mathematical and statistical concepts used in ML, and so a stronger foundation in the field.
- a way to more effectively use systems such as PyTorch or TensorFlow, and customize and fine-tune their models in more sophisticated ways.

Questions?

?

Second part of the lecture

- Overview of probabilistic models
- Maximum likelihood estimation (MLE)
- Sufficient statistics
- Exponential families

Overview of probabilistic models

- Consider a random vector

$$X = (X_1, X_2, \ldots, X_d)$$

that is either observed or partially observed.

- We want to model the relationship between these variables.
- Probabilistic generative models: relate all variables by their joint probability distribution $p(x) = p(x_1, x_2, \ldots, x_d)$.
    - d generally very large; distribution can be complicated

Our objective

Suppose there is a true joint p_* which can be approximated by a model $\mathcal{P}$ (Find $p \approx p_*$ where $p \in \mathcal{P}$).

This course will investigate

- how we should specify a set of distributions $\mathcal{P}$,
- what it means for p to well approximate the true distribution p_*,
- how we can find a reasonable $p \in \mathcal{P}$ efficiently.
- useful modelling assumptions, e.g. conditional independence.

Our objective

Suppose there is a true joint p_* which can be approximated by a model $\mathcal{P}$ (Find $p \approx p_*$ where $p \in \mathcal{P}$).

This course will investigate

- how we should specify a set of distributions $\mathcal{P}$,
- what it means for p to well approximate the true distribution p_*,
- how we can find a reasonable $p \in \mathcal{P}$ efficiently.
- useful modelling assumptions, e.g. conditional independence.

These problem are studied in other statistics courses but here we focus on scalability and autonomy.

A Probabilistic Perspective on ML Tasks

This perspective can model common machine learning tasks, where random variables represent:

- input data x (generally high dimensional),
- discrete outputs ("labels") c (e.g. $\{0, 1\}$),
- or continuous outputs y (e.g. y is daily temperature).

A Probabilistic Perspective on ML Tasks

This perspective can model common machine learning tasks, where random variables represent:

- input data x (generally high dimensional),
- discrete outputs ("labels") c (e.g. $\{0, 1\}$),
- or continuous outputs y (e.g. y is daily temperature).

If we have the joint probability over these random variables, e.g. $p(x, y)$ or $p(x, c)$, we will see later that we can use it for familiar ML tasks:

- **Regression**: $p(y|x) = p(x, y)/p(x) = p(x, y)/\int p(x, y)dy$
- **Classification / Clustering**: $p(c|x) = p(x, c)/\sum_c p(x, c)$

Example: Supervised Classification

We observe pairs of "input data" and "class labels",

$$\{x^{(i)}, c^{(i)}\}_{i=1}^{N} \stackrel{i.i.d.}{\sim} p(x, c).$$

The supervised classification problem will be to learn a distribution over class labels given new input data:

$$p(c|x) = p(x, c) / \sum_{c} p(x, c)$$

Example: Supervised Classification

We observe pairs of "input data" and "class labels",

$$\{x^{(i)}, c^{(i)}\}_{i=1}^{N} \overset{i.i.d.}{\sim} p(x,c).$$

The supervised classification problem will be to learn a distribution over class labels given new input data:

$$p(c|x) = p(x,c) / \sum_{c} p(x,c)$$

- **Discriminative models**: deal only with $p(c|x)$.
- **Generative models**: deal with $p(c,x)$.
  - More complicated, but allows more things (e.g. generating examples from a class: $p(x|c)$)

Observed vs Unobserved Random Variables

Supervised classification: datasets include input data and class labels

- **Supervised Dataset**: $\{x^{(i)}, c^{(i)}\}_{i=1}^{N} \sim p(x,c)$.

  In this case, the class labels are **observed**.

Unsupervised classification: the data still generated from $p(x,c)$ but instead of the pair $\{x^{(i)}, c^{(i)}\}$ we observe only $x^{(i)}$.

What is the probability of observing $x^{(i)}$?

- **Unsupervised Dataset**: $\{x^{(i)}\}_{i=1}^{N} \sim p(x) = \sum_c p(x,c)$.

  The common way to call an unobserved discrete class is "cluster".

Desiderata of Probabilistic Models

In order to learn p_* from data $\{x^{(i)}\}$, we make modelling assumptions:

1. **IID data:** We almost always assume that samples $x^{(i)}$ are i.i.d.
2. **"Parametrized" distributions:** The distribution comes from a parametrized family $\mathcal{P} = \{p(x|\theta) \ : \ \theta \in \Theta\}$. This reduces the complexity of our search space to the complexity of Θ.
   - e.g. $\mathcal{P} = \{p(x|\theta) = N(\theta, 1) \ : \ \theta \in \mathbb{R}\}$, Gaussian distributions with variance 1 and centered around $\theta \in \mathbb{R}$.
   - Θ may still be **very** high dimensional.

Likelihood function

- Let $x^{(i)} \sim p(x|\theta_*)$ for $i = 1, \dots, N$ be i.i.d. random variables.
- The joint distribution of $\mathcal{D} = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ is $p(\mathcal{D}|\theta_*) = \prod_i p(x^{(i)}|\theta_*)$.
- Assume we observe data $\mathcal{D}$ and θ_* is unknown.
- The likelihood function (a function of θ):

$$\mathcal{L}(\theta; \mathcal{D}) = p(\mathcal{D}|\theta) = \prod_i p(x^{(i)}|\theta)$$

Likelihood function

- Let $x^{(i)} \sim p(x|\theta_*)$ for $i = 1, \dots, N$ be i.i.d. random variables.
- The joint distribution of $\mathcal{D} = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ is $p(\mathcal{D}|\theta_*) = \prod_i p(x^{(i)}|\theta_*)$.
- Assume we observe data $\mathcal{D}$ and θ_* is unknown.
- The likelihood function (a function of θ):

$$\mathcal{L}(\theta; \mathcal{D}) = p(\mathcal{D}|\theta) = \prod_i p(x^{(i)}|\theta)$$

- The log-likelihood function:

$$\ell(\theta; \mathcal{D}) = \log \mathcal{L}(\theta; \mathcal{D}) = \sum_i \log p(x^{(i)}|\theta)$$

Note: If x is discrete, $\mathcal{L}(\theta; \mathcal{D})$ is the probability of observing $\mathcal{D}$ if it was generated from $p(x|\theta)$.

Maximum Likelihood Estimation

How to estimate the true parameter θ_*?

- Very intuitive idea: pick parameter values which were most likely to have generated the data

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}}\, \ell(\theta; \mathcal{D}) = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta; \mathcal{D})$$

Maximizing the log-likelihood is equivalent and typically easier.

MLE Example: Bernoulli distribution

- Let $x^{(i)}$ represent the result of the ith coin flip

$$x^{(i)} = 1 \text{ , if heads with probability } \theta \in (0, 1)$$

$$x^{(i)} = 0 \text{ , if tails with probability } (1 - \theta)$$

MLE Example: Bernoulli distribution

- Let $x^{(i)}$ represent the result of the ith coin flip

$$x^{(i)} = 1 \text{ , if heads with probability } \theta \in (0,1)$$

$$x^{(i)} = 0 \text{ , if tails with probability } (1-\theta)$$

- The log-likelihood function is

$$\ell(\theta;\mathcal{D}) = \log p(\mathcal{D}|\theta) = \log \prod_{i=1}^{N} p(x^{(i)}|\theta) = \log \prod_{i=1}^{N} \theta^{x^{(i)}}(1-\theta)^{(1-x^{(i)})}$$

$$= \sum_{i=1}^{N} \left(x^{(i)} \log(\theta) + (1-x^{(i)}) \log(1-\theta) \right)$$

$$= \left(\sum_{i=1}^{N} x^{(i)} \right) \log\theta + \left(N - \sum_{i=1}^{N} x^{(i)} \right) \log(1-\theta)$$

MLE Example: Bernoulli distribution

We maximize

$$\ell(\theta; \mathcal{D}) = \left(\sum_{i=1}^{N} x^{(i)}\right) \log \theta + \left(N - \sum_{i=1}^{N} x^{(i)}\right) \log(1-\theta)$$

by solving

$$\frac{\partial \ell(\theta; \mathcal{D})}{\partial \theta} = \frac{\sum_{i=1}^{N} x^{(i)}}{\theta} - \frac{\left(N - \sum_{i=1}^{N} x^{(i)}\right)}{1-\theta} = 0,$$

which gives

$$\hat{\theta}_{\mathrm{MLE}} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}.$$

(this stationary point can be shown to be a maximum, not required in this course)

Sufficient statistics

- In the previous example, the only aspect of our data that affects the likelihood is the counts $\sum_{i=1}^{N} x^{(i)}$.
- A **sufficient statistic** is a statistic that conveys exactly the same information about the parameter as the entire data.

Sufficient statistics

- In the previous example, the only aspect of our data that affects the likelihood is the counts $\sum_{i=1}^{N} x^{(i)}$.
- A **sufficient statistic** is a statistic that conveys exactly the same information about the parameter as the entire data.
- Fisher-Neyman Factorization Theorem: $T(x)$ is a sufficient statistics for the parameter θ in the parametric model $p(x|\theta)$ iff

  $$p(x|\theta) = h(x)g_\theta(T(x))$$

  for some functions h (that does not depend on θ) and g_θ.

Exponential families

- Density of a member of exponential families is of the form

$$p(x|\eta) \;=\; h(x)\exp\{\eta^\top T(x) - A(\eta)\},$$

  where

$$\begin{aligned} T(x) &: \text{sufficient statistics} \\ \eta &: \text{natural parameter} \\ A(\eta) &: \text{log-partition function} \\ h(x) &: \text{carrying measure} \end{aligned}$$

- Notice that in the exponent, natural parameter interacts with the data only through the sufficient statistics
- Examples: Bernoulli, (multivariate) Gaussian, gamma, exponential, chi-squared, beta, Dirichlet, Poisson, geometric.

Some applications

Exponential families (EFs) have many important applications:

- Many known distributions are EFs.
- Basis for **generalized linear models** (e.g. logistic regression).
- Widely used in multivariate statistics and spatial statistics, e.g., undirected graphical models or the Ising model.
- Many random graph models are exponential families.
- EFs arise as the solution of interesting optimization problems.

The theory of EFs relies heavily on convex analysis.

1-sample example: Bernoulli distribution

We can write this distribution as an exponential family

$$
\begin{aligned}
p(x|\theta) =& \theta^x (1-\theta)^{1-x} \\
=& \exp\{x \log(\theta) + (1-x)\log(1-\theta)\} \\
=& \exp\{x \log(\tfrac{\theta}{1-\theta}) + \log(1-\theta)\}
\end{aligned}
$$

Here,

$$
\begin{aligned}
T(x) =& x \\
\eta =& \log(\tfrac{\theta}{1-\theta}) \\
A(\eta) =& \log(1+e^\eta) \\
h(x) =& 1
\end{aligned}
$$

Notice that $A'(\eta) = \frac{e^\eta}{1+e^\eta} = \theta$ is the mean of $T(X) = X$ and $A''(\eta) = \frac{e^\eta}{(1+e^\eta)^2} = \theta(1-\theta)$ is the variance of X.

Mean of sufficient statistics

Moments of exponential families can be easily computed using the log-partition function. Let $X \sim p(x|\eta)$

$$\mathbb{E}[T(X)] - A'(\eta) = \int T(x) p(x|\eta) dx - A'(\eta)$$

Mean of sufficient statistics

Moments of exponential families can be easily computed using the log-partition function. Let $X \sim p(x|\eta)$

$$\begin{aligned}\mathbb{E}[T(X)] - A'(\eta) &= \int T(x)p(x|\eta)dx - A'(\eta) \\ &= \int \{T(x) - A'(\eta)\}h(x)\exp\{\eta^\top T(x) - A(\eta)\}dx\end{aligned}$$

Mean of sufficient statistics

Moments of exponential families can be easily computed using the log-partition function. Let $X \sim p(x|\eta)$

$$
\begin{aligned}
\mathbb{E}[T(X)] - A'(\eta) &= \int T(x)p(x|\eta)dx - A'(\eta) \\
&= \int \{T(x) - A'(\eta)\}h(x)\exp\{\eta^\top T(x) - A(\eta)\}dx \\
&= \int \frac{d}{d\eta}\Big(h(x)\exp\{\eta^\top T(x) - A(\eta)\}\Big)\,dx
\end{aligned}
$$

Mean of sufficient statistics

Moments of exponential families can be easily computed using the log-partition function. Let $X \sim p(x|\eta)$

$$\begin{aligned}\mathbb{E}[T(X)] - A'(\eta) &= \int T(x)p(x|\eta)dx - A'(\eta) \\ &= \int \{T(x) - A'(\eta)\}h(x)\exp\{\eta^\top T(x) - A(\eta)\}dx \\ &= \int \frac{d}{d\eta}\Big(h(x)\exp\{\eta^\top T(x) - A(\eta)\}\Big)dx \\ &= \frac{d}{d\eta}\int p(x|\eta)dx \\ &= \frac{d}{d\eta}1 = 0.\end{aligned}$$

Thus, we conclude that $\mathbb{E}_\eta[T(X)] = A'(\eta)$.
The variance $\mathrm{var}_\eta(T(X))$ can be computed similarly, and so on.

MLE for general Exponential Families

Recall: $p(x|\eta) \; = \; h(x) \exp\{\eta^\top T(x) - A(\eta)\}$.

After observing data $\mathcal{D}$ with N samples, we write the log-likelihood:

$$\ell(\eta; \mathcal{D}) = \log p(\mathcal{D}; \eta) = \sum_{i=1}^{N} \log h(x^{(i)}) + \eta^\top \sum_{i=1}^{N} T(x^{(i)}) - NA(\eta)$$

For the MLE derivation we solve:

$$\ell'(\eta; \mathcal{D}) \; = \; \sum_{i=1}^{N} T(x^{(i)}) - NA'(\eta) \; = \; 0$$

The MLE for the natural parameters η of a general exponential family:

$$\hat{\eta}_{\text{MLE}} \quad \text{that solves} \quad A'(\hat{\eta}_{\text{MLE}}) = \frac{1}{N} \sum_{i=1}^{N} T(x^{(i)}).$$

Note: This equation may not have an explicit solution but the solution always corresponds to the global maximum.

Summary

- Probabilistic models are our main tool in machine learning.
- We make modelling assumptions (i.i.d., parametric models) for tractability. MLE is one example.
- Exponential families are useful parametric models that provide a general framework.
- More on them when we cover Markov random fields.

Questions?

?