STA 414/2104: Statistical Methods in Machine Learning II

Week 1: Introduction and Preliminaries

Piotr Zwiernik

University of Toronto

Table of contents

1. What is this class about?
2. Administrative details
3. Introduction to statistical learning
4. Overview of probabilistic models
5. Maximum likelihood estimation
6. Exponential families

What is this class about?

This course

- Introduction to **probabilistic** machine learning (PML).
- We introduce many fundamental concepts of machine learning.
- Aimed at advanced undergrad and master level graduate students.
- Homeworks more hands-on but overall the lecture focuses on the theory.
- We will use **a lot of** real analysis, probability, and linear algebra.

Required background

Do I have the appropriate background?

- Linear algebra: vector/matrix manipulations, basic geometric intuitions.
- Calculus: partial derivatives/gradient.
- Probability: common distributions; Bayes Rule.
- Statistics: expectation, variance, covariance, median; maximum likelihood.

Administrative details

Course Information

Course Website: `https://pzwiernik.github.io/sta414/`

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

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

- You received an announcement on Sunday.

We will use Piazza for **discussions**.

- Sign up via quercus or: `https://piazza.com/utoronto.ca/winter2024/sta414`
- Your grade **does not depend on your participation on Piazza**.
- We only allow questions that are related to the course materials/assignments/exams.

Course Information

- This course have two *identical* sections:
  - Section 1: M 2-5pm
  - Section 2: T 6-9pm 😩
- You are welcome to attend either one of the sections.
- 3h = 2h lecture + 1h tutorial
- Instructor office hours are Tuesday 3:30-5pm, UY 9040.
- 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, talking, recording or taking pictures in class is prohibited without the consent of your instructor.
- Lecture slides and notes will be posted on the course webpage. Please do let me know about typos you notice and/or any suggestions you might have.
- This year the Verification of Illness is not required. The absence declaration is considered sufficient to indicate absence. It is student's responsibility to inform the instructor in a timely manner (via e-mail).
- For accessibility services: If you require additional academic accommodations, please contact UofT Accessibility Services as soon as possible, `studentlife.utoronto.ca/as`. No last minute arrangements will be considered.

Literature

Recommended 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
  - 26/27 February (tentative)
  - 2 hours
  - Worth 30% of course mark
- Final Exam (In-Person)
  - $\sim$ 2-3 hours
  - Date and time TBA
  - Worth 40% of course mark
- Exam questions are conceptual/theoretical; no coding.
- Everybody must take the final exam! No exceptions.

Use of generative AI

Homework assignments in this course are designed to reinforce lecture content and provide valuable learning experiences outside of class. While students are encouraged to utilize artificial intelligence tools, including generative AI, as learning aids or for assistance with assignments, it is important to remember that the ultimate responsibility for the submitted work rests with the students.

Grasping complex concepts often involves a layered learning process and intellectual engagement, which cannot be fully substituted by even the most creative use of tools like chat-GPT.

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.
- The schedule of assignments will be posted on the course webpage.
- Assignments should be handed in 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 and Numpy on assignments.

UNIVERSITY OF TORONTO
FACULTY OF ARTS & SCIENCE

LEAD OR JOIN AN RSG

Recognized Study Group

- Meet weekly with up to 8 classmates and make friends
- Increase your understanding of course material
- Prepare for tests and exams
- Build leadership and study skills
- Get CCR recognition

SIDNEY SMITH COMMONS
uoft.me/rsgs

Related Courses

- STA314 and CSC311: Intro ML (we build on these courses)
- **STA414/2104**: This course
- CSC412/2506: Mostly same material
- CSC413: Neural networks and deep learning
- STA302: Linear regression and classical statistics
- CSC2515: Advanced CSC311
- CSC2532: Learning theory - Focus on mathematics of ML
- Various topics and seminar style courses offered at DoSS and DCS

Provisional Calendar (tentative)

- week 1, Jan 8/9:
  - Introduction
  - Probabilistic models (exponential families, MLE)
- week 2, Jan 15/16:
  - Statistical Decision Theory
  - Directed graphical models I (DAGs)
- week 3, Jan 22/23:
  - Directed graphical models II (DAGs)
  - Markov Random Fields
  - Assignment 1 release on Jan 22
- week 4, Jan 29/30:
  - Exact inference
  - Message passing
  - Assignment 1 due on Feb 4

Provisional Calendar (cont'ed)

- week 5, Feb 5/6:
  - Sampling, MCMC
  - Assignment 2 release on Feb 5
- week 6, Feb 12/13:
  - Hidden Markov models
  - Variational inference I
  - Assignment 2 due on Feb 18
- week 7: Reading week
- week 8, Feb 26/27:
  - Midterm exam on Feb 27/28
- week 9, Mar 4/5:
  - Variational inference II
  - EM algorithm

Provisional Calendar (cont'd)

- week 10, Mar 11/12:
  - Bayesian regression
  - Probabilistic PCA
  - Assignment 3 release on Mar 4
- week 11, Mar 18/19:
  - Kernel methods
  - Gaussian processes
- week 12, Mar 25/26:
  - Neural Networks
  - Assignment 3 due on Mar 24
- week 13, Apr 1/2:
  - TBD
- Final Exam

Introduction to statistical learning

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 autonomous agents.
  - Statistics puts more emphasis on interpretability and mathematical rigour; ML puts more emphasis on predictive performance, scalability, and autonomy.
- *Statistical learning* draws heavily from both worlds.

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 are all just special cases of estimating distributions from data: $p(y|x)$, $p(x)$, $p(x, y)$! This is the main focus of this course.

History of machine learning

- 1943 — Perceptron algorithm (implemented as a circuit in 1957)
- 1959 — Arthur Samuel wrote a learning-based checkers program that could defeat him.
- 1969 — Minsky and Papert's book *Perceptrons* (limitations of linear models)
- 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

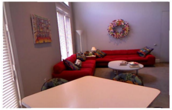
History of machine learning

- 1990s — the "AI Winter", a time of pessimism and low funding
- But looking back, the '90s were also a golden age for ML research
  - Markov chain Monte Carlo, variational inference
  - kernels, support vector machines
  - boosting, convolutional networks
- 2000s — applied AI fields (vision, NLP, etc.) adopted ML
- 2010s — deep learning
  - 2010–2012: 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
  - 2018: Transformer models revolutionized natural language understanding
  - 2020: GPT-3 and DALLE demonstrated impressive text and image generation capabilities

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



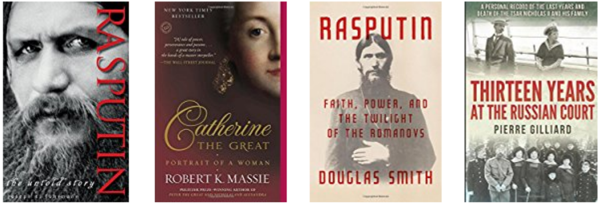
DAQUAR 1553
What is there in front of the sofa?
Ground truth: table
IMG+BOW: table (0.74)
2-VIS+BLSTM: table (0.88)
LSTM: chair (0.47)

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)

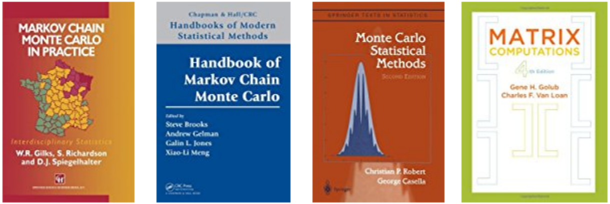
Speech: Speech to text, personal assistants, speaker identification. . .

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

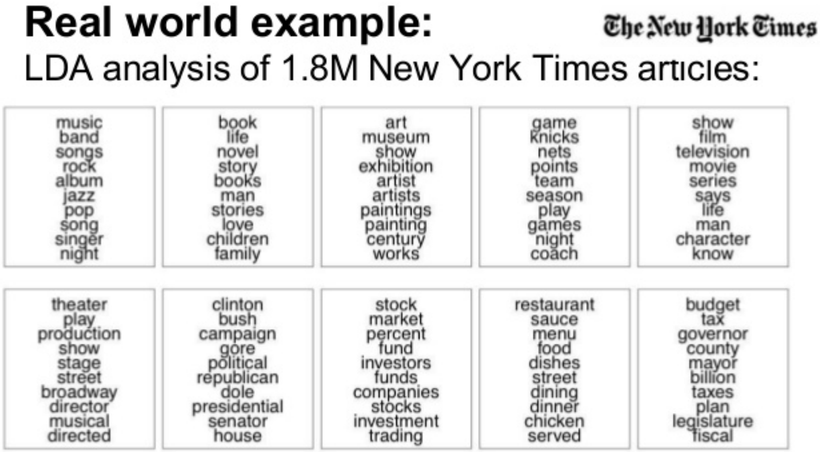
Inspired by your shopping trends



Related to items you've viewed See more



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



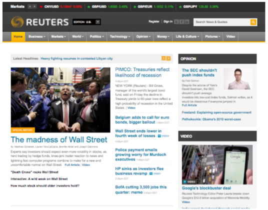
Take a large newswire corpus. A simple model based on the word counts of webpages

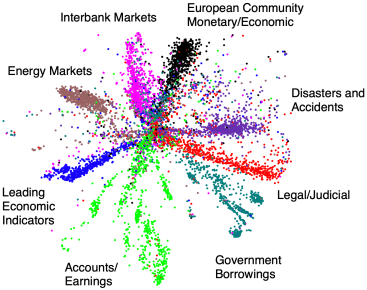
$$P(x) = \frac{1}{Z} \sum_h \exp[x^\top W h]$$

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

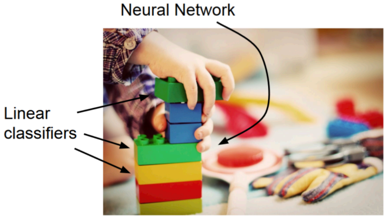


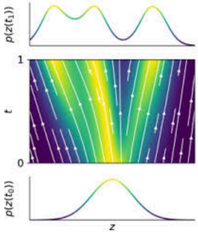


Why this class?

- This class compliments STA 314 with more focus on unsupervised learning.
- We discuss fundamental probabilistic ideas in machine learning.
- 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
- ...

Implementing machine learning systems

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

The theory 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)
- Exponential families

Overview of probabilistic models

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 $p(x) = p(x_1, x_2, \ldots, x_d)$.

Our objective

Suppose the true joint p_* can be approximated by our model $\mathcal{P}$ ($p_* \approx p$ 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

With this perspective, think about common machine learning tasks probabilistically:

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

If we have the joint probability over these random variables, e.g. $p(x, y)$ or $p(x, c)$, 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",

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

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 with $p(c|x)$.
- **Generative models**: deal with $p(c, x)$.

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".

  Possible complication if the number of clusters is unknown.

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.

Maximum likelihood estimation

Likelihood function

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

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

- The log-likelihood function:

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

Natural interpretation in the case when x is discrete

$\mathcal{L}(\theta; \mathcal{D}) =$ 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 typically easier.

MLE Example: Bernoulli distribution

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

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

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

- The log-likelihood function is

$$\begin{aligned}
\ell(\theta;\mathcal{D}) = \log p(\mathcal{D}|\theta) &= \log \prod_{i=1}^{N} \theta^{x^{(i)}} (1-\theta)^{(1-x^{(i)})} \\
&= \sum_{i=1}^{N} \left(x^{(i)} \log(\theta) + (1-x^{(i)}) \log(1-\theta) \right) \\
&= \left(\sum_{i=1}^{N} x^{(i)} \right) \log\theta + \left(N - \sum_{i=1}^{N} x^{(i)} \right) \log(1-\theta)
\end{aligned}$$

MLE Example: Bernoulli distribution

We maximize

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

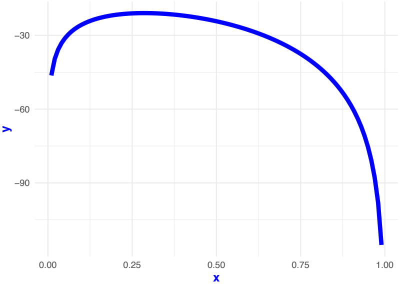
by solving

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

which gives

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

(this stationary point is a maximum)



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 function of the data 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 $p(x|\theta)$ if and only if

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

for some functions h, that does not depend on θ, and g_{θ}.

Exponential families

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: the (multivariate) Gaussian distribution, gamma, exponential, chi-squared, beta, Dirichlet, Poisson, geometric.

Some applications

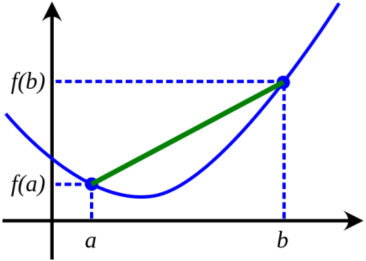
Exponential families have many important applications:

- 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.

A jog through convex optimization

A function $f : \mathbb{R}^d \to \mathbb{R}$ is **convex** if:



for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ and $\lambda \in (0,1)$

$$f((1-\lambda)\mathbf{a} + \lambda\mathbf{b}) \;\leq\; (1-\lambda)f(\mathbf{a}) + \lambda f(\mathbf{b}).$$

Some 1-diml examples: x^2, e^{cx}, $-\log x$, $x \log x$.

If f twice differentiable then:

$$f \text{ convex} \qquad \iff \qquad \nabla^2 f(\mathbf{x}) \text{ positive semi-definite.}$$

Convex Analysis, Section 8.1, [PML1]

May be useful to review some of these concepts.

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 $X \sim p(x|\eta)$ and denote by $A'(\eta) = dA(\eta)/d\eta$

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

Thus, we conclude that $\mathbb{E}_\eta[T(X)] = A'(\eta)$.

The variance $\mathrm{var}_\eta(T(X))$ can be computed similarly.

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 satisfies: $A'(\hat{\eta}_{\text{MLE}}) = \frac{1}{N}\sum_{i=1}^{N} T(x^{(i)})$. 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?

?