

STATISTICAL METHODS FOR MACHINE LEARNING II

© by Xia, Tingfeng

2020 winter term

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International” license.



Preface

This document is consist of notes from lectures and the online course notes that I, personally, find interesting/important. You can find the online course notes on the course website here: <https://probmlcourse.github.io/sta414/>

Contents

1	Lecture 2 - Introduction to Probabilistic Models	2
1.1	Overview	2
1.2	Probabilistic Perspective on ML	2
1.2.1	(Example) Classification	2
1.3	Observed vs Unobserved Random Variables	3
1.3.1	Supervised Dataset	3
1.3.2	Unsupervised Dataset	3
1.3.3	Latent Variables	3
1.4	Operations on Probabilistic Models	4
1.5	Desiderata of Probabilistic Models	4
1.5.1	Fully Dependent Factorization (Chain Rule)	4
1.5.2	Assumptions (Independence)	4
1.6	Likelihood Function	5
1.7	Maximum Likelihood Estimation	5
1.8	Sufficient Statistics	5
1.8.1	Neyman Factorization	5
1.9	Exponential Family	6
1.9.1	(Example) 1-D Gaussian	6

1 Lecture 2 - Introduction to Probabilistic Models

1.1 Overview

We have a random vector in the form $X = (X_1, \dots, X_m)$ which can be *either observed or unobserved*. To approach this in a generative way, we make the so called generative assumption. which is that $X \sim P_{true}(X)$, i.e. there is some true distribution that is behind the scene and our data is from such distribution.

Goal Model a parametric joint distribution $P_\theta(X)$ by learning the parameters. The learning here means we want to find a/the “close”/“best” estimation to our parameter θ . In this course we will investigate the following three problems,

- How to specify the joint, $P_\theta(X)$?
- What does “best”/“close” mean? In some sense we want to find $P_\theta \approx P_{true}$, however P_{true} might also be unknown.
- How to find the best θ ? In this course we will generally rely on gradient methods, so $\nabla_\theta \dots$

1.2 Probabilistic Perspective on ML

With this perspective, we can think about common machine learning tasks differently, where random variables represent:

- X : (high dimensional) input data
- C : discrete label
- Y : continuous target

If we assume our knowledge of the joint of the above three, i.e. we know $P(X, C, Y)$, then we can write our familiar tasks in the following way

- **Regression:**

$$p(Y|X) = \frac{p(X, Y)}{P(X)} = \frac{p(X, Y)}{\int p(X, Y) dY}$$

- **Classification/Clustering:**

$$p(C|X) = \frac{p(X, C)}{\sum_{C'} p(X, C')}$$

1.2.1 (Example) Classification

Suppose we have data of the form $\mathcal{D} = \{(x, c)_i\}_i$. We assume that they came from a certain true distribution, i.e. $\{(x, c)_i\}_i \sim p(X, C)$. Then, the ultimate goal of the ML problem is converted into finding $p(C|X)$. Using Bayes Rule of total probability, we can expand the distribution of interest into

$$p(C|X) = \frac{p(X, C)}{P(X)} = \frac{p(X, C)}{\sum_{C'} p(X, C')}$$

Output Heuristics After we acquire $p(C|X)$ as above, we are one step away from our goal of output the actual prediction c^* . There are three ways that we can do this, namely

- **MLE Estimate** is the most intuitive one, we simply choose

$$c^* = \arg \max_c p(C = c|X)$$

- **Sample Learnt Dist** is another approach which produces non-deterministic results, i.e. we sample $c^* \sim p(C|X)$.
- **Combined** is usually a safe way of doing this. We output

$$(c^*, p(C = c^*|X)) \iff (\text{result, how sure?})$$

As an example, we have a ML algorithm drives a car. In this case, we might want to make decision only when the machine learning model has a certain level of confidence.

1.3 Observed vs Unobserved Random Variables

1.3.1 Supervised Dataset

$$\{x_i, c_i\}_{i=1}^N \sim p(X, C)$$

In such case, the class labels are observed and finding the conditional distribution $p(C|X)$ satisfies the supervised classification problem.

1.3.2 Unsupervised Dataset

$$\{x_i\}_{i=1}^N \sim p(X, C)$$

Still under the generative assumption, where we assume that there is some underlying distribution for our dataset. Further, we assume that the distribution of data is related to the class labels for the data points even though the class labels are never observed. **A common way to refer to an unobserved discrete class label is “cluster”**. However, in this case, our final goal of classification is still $p(C|X)$ ¹.

1.3.3 Latent Variables

Further, like clusters, introducing assumptions about unobserved variables is a powerful modelling tool. We will make use of this by modelling variables which are never observed in the dataset, called latent or hidden variables. By introducing and modelling latent variables, we will be able to naturally describe and capture abstract features of our input data.

¹Might be helpful to think of Gaussian Mixture Models

1.4 Operations on Probabilistic Models

- **Generate Data:** Sample from the model.
- **Estimate Likelihood:** When all variables are either observed or marginalized the result is a single real number which is the probability of the all variables taking on those specific values.
- **Inference:** Compute the expected value of some variables given others which are either observed or marginalized.
- **Learning:** Set the parameters of the joint distribution given some observed data to maximize the probability of the observed data.

1.5 Desiderata of Probabilistic Models

We have to desires for the joint distribution to learn, namely

- The marginal and conditional distribution can be computed efficiently
- The representation of the joint distribution should be compact. This is especially important when we are dealing with joint distributions over many variables.

In general, total joint distribution are too large to specify and would require an insane amount of data to fit even we wanted to. Thus, we need modelling assumptions.

1.5.1 Fully Dependent Factorization (Chain Rule)

Suppose we have sample space defined such that $|T| = 2$, $|W| = 3$, and $|M| = 4$. Then the total joint distribution could be expanding using the chain rule as (*note: not unique*)

$$P_{\theta}(T, W, M) = P(T)P(W|T)P(M|T, W)$$

which requires² $|\theta| = (2 - 1) + (3 - 1) \times 2 + (4 - 1) \times 2 \times 3 = 23$ parameters in total to specify.

1.5.2 Assumptions (Independence)

Introducing assumptions results in

- a less expressive model
- $|\theta|$ (usually, much) smaller

which can be bad sometimes and is a trade-off that we as modellers have to deal with.

(Example) Fully Independent We assume that $T \perp\!\!\!\perp W \perp\!\!\!\perp M$, then by definition we know, for example, $P(W|T) = P(W)$. Then

$$\begin{aligned} P_{\theta}(T, W, M) &= P(T)P(W|T)P(M|T, W) \\ &= P(T)P(W)P(M) \end{aligned}$$

which requires only $|\theta| = (2 - 1) + (3 - 1) + (4 - 1) = 6$ to fit.

²Here the bars mean “cardinality” rather than vector norm

1.6 Likelihood Function

For some observed data X , the likelihood describes the likeliness of the data under then distribution with parameter θ .

$$L(\theta) = p(X|\theta)$$

In general, we prefer to deal with the log likelihood function, which is defined as

$$\ell(\theta; X) = \log L(\theta) = \log p(x|\theta)$$

1.7 Maximum Likelihood Estimation

The idea is to find

$$\hat{\theta}_{MLE} := \arg \max_{\theta} \ell(\theta; \mathcal{D})$$

In the case of i.i.d, we can re-write as

$$\begin{aligned} \hat{\theta}_{MLE} &= \arg \max_{\theta} \ell(\theta; \mathcal{D}) \\ &= \arg \max_{\theta} \log \prod_m p(x^{(m)}|\theta) \\ &= \arg \max_{\theta} \sum_m \log p(x^{(m)}|\theta) \end{aligned}$$

1.8 Sufficient Statistics

(Definition) Statistic A statistic is a possibly vector valued *deterministic* function of a set of random variables.

(Definition) Sufficient Statistic is a statistic that conveys exactly the same information about the data generating process that created the data as the entire data itself.³ In formal language, Sufficient Statistic ($T(X)$) for X could be defined as

$$T(X^{(1)}) = T(X^{(2)}) \implies L(\theta; X^1) = L(\theta; X^2), \quad \forall \theta$$

alternatively, we can define it as

$$P(\theta|T(X)) = P(\theta|X); \quad \text{i.e. data doesn't give further info}$$

1.8.1 Neyman Factorization

Equivalently to the above, we can write

$$P(\theta|T(X)) = h(x, T(x))g(T(x), \theta)$$

³Could also be interpreted as “summarize the data with respect to the likelihood”

1.9 Exponential Family

Factorizes as

$$\begin{aligned} p(x|\eta) &= h(x) \exp \{ \eta^T T(x) - g(\eta) \} \\ &= h(x) g(\eta) \exp \{ \eta^T T(x) \} \end{aligned}$$

1.9.1 (Example) 1-D Gaussian

$$\begin{aligned} p(x|\theta) = \mathcal{N}(x|\mu, \sigma) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{1}{2\sigma^2} (x - \mu)^2 \right) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{1}{2\sigma^2} (x^2 - 2x\mu + \mu^2) \right) \\ &= \underbrace{\frac{1}{\sqrt{2\pi\sigma}} \exp \left(\frac{-\mu^2}{2\sigma^2} \right)}_{\log A(\eta)} \exp \left(\underbrace{\begin{bmatrix} \frac{\mu}{\sigma^2} & \frac{-1}{2\sigma^2} \end{bmatrix}}_{\eta} \underbrace{\begin{bmatrix} x \\ x^2 \end{bmatrix}}_{T(X)} \right) \end{aligned}$$