Chapter 2

Probabilistic Modelling

2.1 Introduction

Probabilistic modelling is a set of techniques that leverage probability distributions and random variables to posit, test and refine hypothesis about the behaviour of systems. Given observations of a system, the task of probabilistic modelling normally boils down to finding a probability distribution which:

- Is consistent with the observed data;
- Is consistent with **new**, previously unobserved data, originated in the same system.

This probability distribution is commonly called the *model*. A good model will be a good *emulator* of the true generative process that originated the observed data. In loose terms, this can be summarized as:

$$\text{data} \sim p(\text{data}|\text{hypothesis}^*), \tag{2.1}$$

where hypothesis* is the optimal hypothesis.

Via Bayes' Law, we can write:

$$p(\text{hypothesis}|\text{data}) = \frac{p(\text{data}|\text{hypothesis})p(\text{hypothesis})}{p(\text{data})} \tag{2.2}$$

In practice, a modeller will search for an hypothesis that maximizes (some form, or approximation of) this expression. For simpler problems, this search happens in closed-form, i.e., there is an expression to compute the optimal hypothesis, given data. However, for most real-world problems there is no closed-form solution, and the modeller has to resort to algorithms and approximations, and will only be able to find **local** optima for the above expression, in most cases.

It's also worth noting that there are effectively infinite candidate distributions - each one an hypothesis for how the system at hand generates data. It is common to make use of domain knowledge and assume the true system has a certain intrinsic structure and form, and to use these assumptions to constrain the space of candidate hypothesis. Assumptions about structure usually translate to conditional

independence claims between some or all of the observed variables; assumptions about form translate into the use of specific parametric families to govern some or all of the observed variables. These assumptions are commonly connected between themselves (for example, when conjugate likelihood-prior pairs are used).

When parametric forms are used, an hypothesis is uniquely defined by the set of parameters it requires - commonly called θ[1].

2.2 Model Complexity

Intuitively, the size of θ is deeply connected with the expressiveness of the distribution. In practice, this translates to the observation that if we make the model[2] expressive enough, it can fit the observed data arbitrarily well. Naïvely, this would be a desirable characteristic to exploit - it's always possible to increase the likelihood by adding parameters to the model. However, increasing model complexity normally comes at the expense of generalization. This phenomenon is commonly referred to as *overfitting*, and there are several angles from which to explain it and interpret it. Namely:

- The classical perspective is that of the bias-variance tradeoff. To understand this, consider the concept of an Hypothesis Class - a set of hypotheses in which, via some procedure, the modeller will search for an hypothesis that is consistent with the observed data, and is expected to generalize to unseen data. Said procedure is what is normally referred to as *fitting* the model to the data. In the case of parametric models, the set of models of a given parametric form, with a parameter-vector of a certain fixed size, is an example of an Hypothesis Class. Intuitively, a more complex Hypothesis Class is more likely to contain the true hypothesis (or a good approximation to it). However, the more complex the Hypothesis Class, the larger the search-space - the higher the number of candidate hypothesis. In this sense, an increase in the size of the search-space often translates into an increase of the sensitivity to the problem variables (in the case of learning and inference, this means sensitivity to initialization and to the data used to fit). Conversely, a simpler model will constitute a smaller search-space, hence the search procedure will be less sensitive to initalization and problem variables. However, the true hypothesis (or a good approximation to it) is less likely to be contained in it - precisely because it is a smaller Hypothesis Class. The bias-variance tradeoff is a summary of these observations: a highly complex model is potentially able to achieve a low error on observed data (low bias), tend to be extremely sensible to small variations on its input (high variance). Conversely, a simpler model will be more robust to variations on its input (low variance), but won't have the same modelling capacity and will produce larger errors (high bias).

- Andrey Kolmogorov's and Gregory Chaitin's ideas on Algorithmic Information Theory [1], and Kol-

[1] For the type of models and problems dealt with in this work, I will assume θ is finite, but it's worth noting that there are models for which the size of θ *grows* with the dataset size. These are called non-parametric models. They come with their own advantages and disadvantages, which are out of the scope of this work.

[2] Throughout this work I will be using the words *model* and *distribution* almost interchangeably, making it clear when context isn't enough.

mogorov complexity are another useful lens through which to regard this question. Consider that data are measurements of phenomena. Modelling is concerned with finding the laws that explain/govern these phenomena. Intuitively, if the laws are as complex as the data they intend to explain, they aren't explaining anything. AIT formalizes this notion by borrowing the concept of *program* to define the generative process by which observed data comes to existence. The complexity of a dataset is then easy to define: it is the size of the **smallest**[3] program that generates the observed data. And the appropriate unit with which to measure the size of a program - and, as we've now seen, the complexity of a dataset - is bits[4]. The parallel between these ideas and the question of overfitting is thus easy to make: a program (or a model and its parameter vector) is useful if it *compresses* the data, intuitively because to do so it leverages the patterns therein, which are the object of interest in the modelling task.

Both of these lines of reasoning make clear that there is a certain balance in complexity that a good model has to achieve: it should be parsimonious enough that it won't overfit, but flexible enough that it is able to properly explain the observed data. There are strategies to make this mathematically objective. Some of those methods are the Bayesian Information Criterion, the Akaike Information Criterion and the Minimum Description Length.

2.3 Structure and Latent Variables

In some cases, one might want to leverage some available domain knowledge. This often translates into assuming that there is some latent structure in the data. This structure is encoded into latent variables and their influence over the observable variables.

In this scenario, we become interested in the distribution given by $p(x, z, \theta_x, \theta_z)$, where z is the latent variable, θ_x is the parameter vector for the distribution over $\mathcal{X}$, and θ_z is the parameter vector for the distribution over $\mathcal{Z}$.

For structure and latent variables to be useful we normally make the additional assumption that we have the ability of factoring that distribution in ways that make it tractable. One common factorization is:

$$p(x, z, \theta) = p(x|z, \theta_x)p(z|\theta_z)p(\theta_x)p(\theta_z) \tag{2.3}$$

,

2.4 Approximate Inference

For simplicity, let us consider θ as part of the latent variables z. This means that the model is simply written as the joint distribution: $p(x, z)$. *Inference*[5] is the task of finding the most probable z after having

[3] Note the emphasis on "smallest" - this is because any program can be made arbitrarily redundant, and thus arbitrarily large.

[4] Or the basic unit of memory of the computer where the data generating program would run

[5] If we hadn't collapsed θ into z and were instead handling separately, we would call **inference** to the task of finding z and **learning** to the task of finding θ

observed x . Specifically, the goal is to find the posterior distribution of z, given x, i.e.: $p(z|x)$.

Recall Bayes' Law:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{2.4}$$

$$= \frac{p(x|z)p(z)}{\int p(x|z')p(z')dz'} \tag{2.5}$$

For the vast majory of cases, the integral on the denominator will be intractable. To overcome this difficulty we normal resort to two families of methods: Monte-Carlo methods, and Variational methods.

2.4.1 Monte-Carlo Methods

Monte-Carlo methods work by using sampling techniques to approximate the intractable integral. The most powerful subclass of these methods is called Markov-chain Monte-Carlo. Its approach consists of devising a scheme that allows for sampling from a distribution close to the one of interest. It accomplishes this by defining a Markov-Chain whose transition function is guaranteed to make it converge asymptotically to the distribution of interest, given some constraints (ergodicity...) (TODO: explain more)

2.4.2 Variational Methods

Variational methods work by turning the problem of integration into one of optimization. They propose a family of parametric distributions, and then optimize the parameters so as to minimize the "distance" between the approximate (normally called "variational") distribution and the distribution of interest.

There are two ways to derive the most commonly used objective function for this problem.

Kullback-Leibler Divergence

The Kullback-Leibler divergence is a measure[6] of the distance between two probability distributions p, and q. It is given by:

$$KL(q||p) = \int q \log \frac{q}{p} \tag{2.6}$$

In the setting of inference, p is the posterior $p(z|x)$ and q is a distribution in some parametric family, with parameters ϕ, i.e., $q(z;\phi)$. However, it's clear that we can't compute the Kullback-Leibler directly, because it requires the knowledge of both the distributions, and finding $p(z|x)$ is precisely the task at hand. Let us expand the KL divergence expression:

[6]Note that the KL divergence isn't symmetric and as such I haven't called it a *metric*

$$KL(q||p) = \int q(z)(\log q(x) - \log p(z|x))dz \tag{2.7}$$
$$= \int q(z)(\log q(z) - (\log p(x,z) - \log p(x)))dz \tag{2.8}$$
$$= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(x,z)] + \mathbb{E}_q[\log p(x)] \tag{2.9}$$
$$= \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(x,z)] + \log p(x) \tag{2.10}$$

The last term is constant w.r.t $q(z)$. In that sense, for a fixed $p(x)$, minimizing the KL divergence is equivalent to minimizing

$$\mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(x,z)] \tag{2.11}$$

, which is equivalent to maximizing

$$\mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)] \tag{2.12}$$

. This quantity is commonly refered to as ELBO - Expectation Lower BOund. It can be rewritten as:

$$ELBO(q) = \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)] \tag{2.13}$$
$$= \mathbb{E}_q[\log p(x|z)] + \mathbb{E}_q[\log p(x|z)] - \mathbb{E}_q[\log q(z)] \tag{2.14}$$

In this form, each term of the ELBO has an easily interpretable role:

- $\mathbb{E}_q[\log p(x|z)]$ tries to maximize the conditional likelihood of x. That can be seen as assigning high probability mass to values of z that *explain* x well.

- $\mathbb{E}_q[\log p(x|z)]$ is the symmetric of the crossentropy between $q(z)$ and $p(x|z)$. Maximizing this quantity is equivalent of maximizing that crossentropy. This can be regarded as a regularizer that discourages $q(z)$ of being too different from the prior $p(z)$.

- $-\mathbb{E}_q[\log q(z)]$ is the entropy of $q(z)$. Maximizing this term incentivizes the probability mass of $q(z)$ to be spread out: another form of regularization.

A lower bound on $\log p(x)$

Another way of approaching the intractable posterior is to start by stating that our inherent goal is to maximize $p(x)$, or equivalently $\log p(x)$. Given that, consider the following:

$$\log p(x) = \log \int p(x, z) dz \tag{2.15}$$

$$= \log \int q(z) \frac{p(x, z)}{q(z)} dz \tag{2.16}$$

$$= \log \mathbb{E}_q [\frac{p(x, z)}{q(z)}] \tag{2.17}$$

$$\geq \mathbb{E}_q [\log \frac{p(x, z)}{q(z)}] \tag{2.18}$$

$$\geq \mathbb{E}_q [\log p(x, z)] - \mathbb{E}_q [q(z)] \tag{2.19}$$

To understand this derivation, consider Jensen's inequality, given (in one of its forms) by:

$$\phi(\mathbb{E}[X]) \leq \mathbb{E}[\phi(X)], \tag{2.20}$$

where $\phi(.)$ is a convex function.

If $\xi(.)$ is a concave function, then $-\xi(.)$ is a convex function, and we obtain the reverse inequality (substituting $\phi(.)$ with $-\xi(.)$ in the inequality in 2.20):

$$-\xi(\mathbb{E}[X]) \leq \mathbb{E}[-\xi(X)] \tag{2.21}$$

$$\xi(\mathbb{E}[X]) \geq \mathbb{E}[\xi(X)] \tag{2.22}$$

This form is the most useful for us, since $\log$ is a concave function. Using this, the step between 2.17 and 2.18 is made obvious.

Note that the right-hand side of 2.19 is the same quantity we arrived at in 2.12, and that it is a lower-bound on the quantity we want to maximize, and so we want to maximize it. It's worth noting that when $q(z) = p(z|x)$, the bound is tight.