Lecture 1: Probabilistic modeling and parameter estimation

Luca Ambrogioni

In this lecture we will learn how to define probabilistic models and estimate their parameters. Probabilistic models are extremely important tools in most branches of science and engineering. For example, modern weather forecasts are probabilistic models that integrate knowledge of the atmospheric dynamics with various sources of uncertainty in order to make a probabilistic estimates of quantities of interest such as temperature, humidity and precipitation in a given location. The laws of atmospheric dynamics are given in the form of a system of partial differential equations coupling variables such as air density, humidity and wind velocity. These equations tell us how to propagate a "snapshot" of the current state of the atmosphere (e.g. initial conditions) to the future. On the other hand, he biggest source of uncertainty comes from the sparse and imperfect measurement of the initial condition together with the chaotic nature of the equations (small initial changes are amplified into big future differences). This example reveal a general principle of probabilistic models: they integrate knowledge of what is known about a system with a model of what is not known.

The atmospheric models used in meteorology are extremely complicated. On the other hand, probabilistic models can often be highly simplified while retaining most of their usefulness. For example, we could model a coin toss using a Newtonian rigid body simulator that captures the complex dynamics of the physical system. However, it is more practical and almost as useful to model the same system in a much simple way by assigning a probability θ to the coin to land on head and $1 - \theta$ to land on tail.

1 Probabilistic models

Mathematically, a probabilistic model is defined by an array of random variables $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ together with a joint probability distribution $p(\boldsymbol{x})$. For example, the model of a coin toss we discussed above is defined by a binary variable $b \in (0, 1)$ and a Bernoulli distribution:

$$p(b; \theta) = \theta^b (1 - \theta)^{1-b} \ . \tag{1}$$

So what is the difference between a probabilistic model and a probability distribution? A distribution is a mathematical abstraction that can be applied to many different settings, a probabilistic model is the use of a distribution to describe the probabilistic behavior of a real-world phenomenon such as a coin toss. Note that the same Bernoulli distribution can be used as a model of many more real-world phenomena such as the determination of sex in a new child and the probability of getting a job you applied for.

The coin toss model we defined above is characterized by a quantity θ (i.e. the probability of getting a head). We will refer to θ as a parameter of the model. The value of parameters can be known a priori or can be estimated from data.

1.1 Example 1: Random walk down Wall Street

The dynamics of most real-world phenomena is the result of an extremely complex interaction between a large amount of entities. For example, the dynamics of stock prices in finance is the result of hundreds of thousands of investors and traders attempting to exchange the stock for money. The price of a stock is the meeting point between how much buyers are willing to pay and sellers are willing to sell. A detailed model of the price of a stock would then involve the a model of each buyer and seller which in turn are very complex entities (human beings!) each with a different set of beliefs and motivations and inextricably embedded in society at large. However, we can capture some qualitative characteristic of the dynamics of a stock using a very simple probabilistic model which simply say that the price moves at random!

A stock price is an example of time series. The price of the stock is defined at a range of time indices $1, \ldots, t, \ldots, T$ and the price at a given time point depends (in a probabilistic sense) on the price at previous time points. In our simple model, the price of a stock at the next time point is equal to the current price x_t plus a random movement ϵ_t:

$$x_{t+1} = x_t + \epsilon_t \tag{2}$$

$$\epsilon_t \sim \mathcal{N}(\epsilon_t; 0, \sigma^2) \ , \tag{3}$$

where we assume the random movement to follow a normal distribution with zero mean and variance σ^2. Note that this imples that the probability of the next price given the previous is given by $p(x_{t+1} \mid x_t; \sigma^2) = \mathcal{N}(x_{t+1} \mid x_t, \sigma^2)$. If we assume that the price today at $t = 0$ is x_0, we can now write down a probabilistic model for the next T days:

$$\begin{aligned} p(\boldsymbol{x}_{1:T}; \sigma, x_0) &= \prod_{t=1}^{T-1} p(x_{t+1} \mid x_t; \sigma) && (4) \\ &= \prod_{t=1}^{T-1} \mathcal{N}(x_{t+1}; x_t, \sigma^2) && (5) \\ &= \prod_{t=1}^{T-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_{t+1}-x_t)^2/2\sigma^2} \ . && (6) \end{aligned}$$

We can now use the model to make probabilistic predictions about the future!

in this case, we will first do this by mathematical derivation and then using TensorFlow probability.

1.1.1 A mathematical analysis of our stock market model

The joint probability in Eq. 4 provide a full mathematical description of our model of the stock price. However, it can still take a lot of work to use the model to answer the questions we are interested in. For example, we are often interested in the probability of the price at a given time T in the future given the current price x_0. in other words, Eq. 4 gives the joint probability of all intermediate time points while we are interested in the probability $p(x_T \mid x_0)$ that "average over" all the intermediate prices. Formally, we can obtain this distribution by integrating all the intermediate variables:

$$p(x_T \mid \sigma, x_0) = \int_{-\infty}^{\infty} p(x_1, \ldots, x_{T-1}, x_T; \sigma, x_0) \mathrm{d}x_1 \ldots \mathrm{d}x_{T-1} \ . \tag{7}$$

However, solving this integral is challenging and it is not possible in most more complex models. In this case the integral can be solved analytically but there is a much simpler way to get the same result. By applying our transition model in Eq. 2 recursively we get a formula for x_T:

$$\begin{aligned} x_T &= x_{T-1} + \epsilon_T \\ &= x_{T-2} + \epsilon_{T-1} + \epsilon_T \\ &= x_0 + \sum_{t=1}^{T} \epsilon_t \ . \end{aligned} \tag{8}$$

All the increment variables ϵ_t are statistically independent normal variables with mean 0 and variance σ^2. This imply that the summed variable $\sum_{t=1}^{T} \epsilon_t$ is itself normal with mean $\sum_{t=1}^{T} 0 = 0$ and variance $\sum_{t=1}^{T} \sigma^2 = (T-1)\sigma^2$ (check any introductory text in probability and statistics if you are unsure about those results). We can therefore conclude that

$$p(x_T \mid \sigma, x_0) = \mathcal{N}(x_T; x_0, T\sigma^2) \ . \tag{9}$$

Therefore, our model predicts that the mean of the price stays constant at the initial value while the standard deviation grows with the square root of the time T. The growth of the standard deviation reflects our reduced certainty as we try to extrapolate our knowledge far in the future.

1.2 Example 2: Physical motion

Physics provides mathematical models of the physical world. While the fundamental models are often deterministic, when we apply a physical model to the real world we often need to incorporate randomness so account for the physical degrees of freedom that we are not modeling explicitly.

Consider a simple model of a dust particle with mass m. We denote the vertical position of the particle at time t as $x(t)$. Its dynamic is given by Newton's equation of motion:

$$m\dot{v}(t) = F_t \ . \tag{10}$$

where the dot denotes a time derivative and $v(t) = \dot{x}(t)$ is the velocity of the particle. The main force acting on the particle is a constant gravitational pull $-g$ and the air resistances which is proportional to the velocity. Furthermore, since the mass of the particle is very small, its velocity is influenced by a large number of collisions with molecules in the air. hence, the total force acting on the particle is

$$F_t = -mg - a + v(t)\epsilon_t \ . \tag{11}$$

where g is the g-force, a determines the amount of air resistance and ϵ_t is a random Gaussian force with mean zero and variance σ^2. We can summarize this model as a system of differential equations:

$$\begin{aligned} m\dot{v}(t) &= -mg - av(t) + \epsilon_t \\ \dot{x}(t) &= v(t) \ . \end{aligned} \tag{12}$$

Differential equations with random forces are called Langevin equations in physics. The simplest way to deal with them is to discretize the equation using a small finite time step dt:

$$\begin{aligned} v_{t+1} &\sim \mathcal{N}\left(v_{t+1}; v_t - g + \frac{av_t}{m}\mathrm{dt}, \frac{\sigma^2}{m^2}\mathrm{dt}\right) \\ x_{t+1} &= x_t + v(t)\mathrm{d}t \ . \end{aligned} \tag{13}$$

This discretization scheme is called Euler-Maruyama. The crucial insight is that the variance of the distribution scales with the step dt so that the standard deviation scales with $\sqrt{\mathrm{d}t}$. This is analogous to what we saw in the previous section where we found out the the variance of the random walk scales with T.

Note that the transition probabilities in the velocity process have the form:

$$p(v_{t+1} \mid v_t) = \mathcal{N}\left(v_{t+1}; \alpha v_t + \beta, \sigma^2 \mathrm{d}t\right) \tag{14}$$

where α and β are numbers. In words, the mean at the next time point is a linear function of the value in the previous time point. Processes of this form are called autoregressive since each new time step is a "linear regression" of the previous. Autoregressive processes are very important in signal processing, neuroscience, finance and econometrics.

1.3 General time series models

We can now generalize the approach we used in the previous examples to discuss general time series models. In general, the variable x_t can depend on the values of the variable at all previous time points. This can be written as follows:

$$p(x_1, \ldots, x_T) = p(x_0) \prod_{t=0}^{T-1} p(x_{t+1} \mid x_0, \ldots, x_t) \ . \tag{15}$$

Models of this form very popular in the deep learning literature. For example, natural language models based on LSTM, GRUs or transformers all have this probabilistic form. In a language model, the variable x_t is a one-hot-encoded vector that represents a single word. The conditional probabilities $p(x_{t+1} \mid x_0, \dots, x_t)$ are the output of a recurrent neural architecture that takes the first t words as input and output a probability distribution over the next word through a softmax layer.

Markov models are very important special case of time series model where each variable solely depend on the variable at the previous time point:

$$p(x_1, \dots, x_T) = p(x_0) \prod_{t=0}^{T-1} p(x_{t+1} \mid x_t) \ . \tag{16}$$

In a Markov model, the future depends on the past only through the present. All are fundamental laws of physics have this property, however the Markov property often does not apply in non-fundamental model. For example, a Markov model of language would have very low performance since the probability of the next word should depend on many past words.

1.4 Structures of conditional independence

Two variables x and y are said to be statistically independent when their joint probability can be factorized into a product of their marginal:

$$p(x, y) = p(x)p(y) \ . \tag{17}$$

Perhaps more intuitively, this implies that

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

In words, knowing the value of y does not provide any information concerning the value of x.

Two variables x and y are said to be conditionally independent given a third variable z when:

$$p(x, y \mid z) = p(x \mid z)p(y \mid z) \ . \tag{18}$$

Note that two conditionally independent variables are not necessarily independent since they can be coupled through the variable z.

A probabilistic model is defined by a structure of conditional Independence. For example, in the Markov timeseries models we discussed above, a variable x_t is statistically independent to all past variables x_n once we condition on the immediate past x_{t-1}. In other words, knowing the value of x_n does not provide any further information on the value of x_t if we already know the value of x_{t-1}. Mathematically, we can express this conditional independence as follows:

$$p(x_t \mid x_{t-1}.x_n) = p(x_t \mid x_{t-1}) \tag{19}$$

if $n < t - 1$.

1.5 Statistical models and machine learning

In the previous examples, we constructed a model starting from detailed domain knowledge about the field of application. However, very often in data science we need to analyze data sources that whose dynamics is very poorly understood. How can we model what we do not know? A possibility is to start with a very flexible model with many parameters which we can adjust to fit the data. Examples of this kind are very common in statistics, I therefore refer to them as statistical models. For example, we can model the relation between the price y of a house and a series of factors $\boldsymbol{x}$ such as location and size using a linear model:

$$p(y \mid \boldsymbol{x}; W, b) = \mathcal{N}(y; W\boldsymbol{x} + b, \sigma^2) \ . \tag{20}$$

where W is a matrix of regression coefficients and $\boldsymbol{b}$ a vector known as bias or intercept. Note that we did not start with any specific assumption about the complexity of the house market, we simply chose a simple and convenient model that we are going to adapt by tuning the parameters W and $\boldsymbol{b}$.

This is the starting point of machine learning. Linear models are constrained objects with limited flexibility. In machine learning, we tend to use more complex and highly parameterized model that can "learn" from the data by setting their parameters. For example, we can replace the linear model with a D layered deep network parameterized by D matrices and bias vectors:

$$p(y \mid \boldsymbol{x}; W_1, \boldsymbol{b}_1, \dots, W_D, \boldsymbol{b}_T) = \mathcal{N}(y; \mathrm{DNN}(\boldsymbol{x}; W_1, b1, \dots, W_D, \boldsymbol{b}_T), \sigma^2) \ . \tag{21}$$

The big advantage of this approach is that it can be used in almost any situation without much thinking. However, we pay two prices for our "laziness". First, highly parameterized statistical models can overfit the data and become very poor predictor of future events. Second, the parameters do not have any real meaning and therefore the model can become very difficult to interpret. These shortcoming is especially problematic in applications such as medicine where possible undetected mistakes have very serious consequences.

1.6 Combining machine learning with explicit modeling

There ought not to be a clear separation between convectional probabilistic model and statistical/machine learning models. In fact, statistical and explicit modeling can be nicely integrated so to leverage our knowledge of a system while keeping the machine learning flexibility to account for the parts of the model where our prior knowledge is sparse.

For example, we can re-analyze our model of physical motion. The model had two important components: 1) the Newton's equations of motion and 2) a model of the forces acting on the dust particle. Note that we are much more confident about component 1 than about component 2. Therefore, we could make a more flexible model where we keep the Newton equations but we learn the structure of the force from some observed trajectory using a networks:

$$m\dot{v}(t) = F(x(t), v(t), t) \tag{22}$$

$$F(x(t), v(t), t) = \mathrm{DNN}_1(x(t), v(t), t; \theta_1) + \mathrm{DNN}_2(x(t), v(t), t; \theta_2)\epsilon_t \tag{23}$$

where DNN_1 and DNN_2 ar deep networks parameterized by the set of parameters (weights and biases) θ_1 and θ_2 respectively. Note that DNN_1 models the deterministic force while DNN_2 models the scale of the force arising from random collisions. We can now discretize this model using again the Euler-Maruyama scheme. The result is a recurrent neural network architecture that can be trained using standard SGD but that also embeds our physical knowledge about the system (i.e. Newton's law). The most important aim of this course is to give you the tools to combine these two forms of modeling in a unified deep learning framework.

2 Parameter estimation and maximum likelihood

So far we learn how to construct and analyze probabilistic models. The models are characterized by a set of parameters. Sometimes the value of these parameters can be obtained from existing scientific literature or other source of prior knowledge. However this explicit approach is rarely possible especially in highly parameterized statistical models. It is therefore important to have a general strategy to estimate the parameters from the measured data.

Fortunately, the statistical theory gives us a very powerful tool: the maximum likelihood principle. Consider a set of data points $\{\boldsymbol{x}_j\}$ measured from a real-world system that we wish to describe with the probabilistic model $p(\boldsymbol{x}; \theta)$. The maximum likelihood principle says that we can set the parameters theta by maximizing the model (log-)probability of the data given the parameters. Specifically, consider a set of j independently sampled data points $\boldsymbol{x}^{(j)}$. We can write a loss function as the negative log-likelihood:

$$\mathcal{L}(\theta) = \sum_{j}^{J} \log p(\boldsymbol{x}^{(j)}; \theta) \ . \tag{24}$$

where we are summing over the likelihood of the individual datapoints since we assumed that they are sampled independently.

For example, the likelihood of the random walk model of the stock price is given in Eq. 4. Using this formula, we can get a loss function for the variance $\nu = \sigma^2$:

$$\mathcal{L}(\nu) = \sum_{j}^{J} \sum_{t=1}^{T-1} (x_{t+1}^{j} - x_{t}^{j})^2 / 2\nu + TJ \log\left(2\pi\nu\right) \ . \tag{25}$$

In this example, we can obtain the maximum likelihood estimate of the variance of the stock movements by using some simple calculus and algebra:

$$\nabla_\nu \mathcal{L}(\nu) = 0 \Longrightarrow \nu = \frac{1}{JT} \sum_{j}^{J} \sum_{t=1}^{T-1} (x_{t+1}^j - x_t^j)^2 \ . \tag{26}$$

Note that the results makes a lot of sense and it would likely be used by smart data scientist even without any knowledge of the maximum likelihood principle. The estimated variance is simply the empirical variance of the increments estimated from the observed time series.

2.1 Stochastic gradient descent

In most cases it is not possible to maximize the log-likelihood exactly. Fortunately, there is a general strategy to learn the parameters of a general and arbitrarily complicated differentiable model given the data. A probabilistic model is said to be differentiable when the probability $p(\boldsymbol{x}; \theta)$ is differentiable with respect to the parameter θ. In differentiable models, we can minimize the loss by gradient descent:

$$\theta_{n+1} = \theta_n - \eta \nabla \mathcal{L}(\theta_n) \ , \tag{27}$$

where η is a small learning rate. However, when the dataset is large it is unfeasible to compute the gradient over all the data points. It it therefore usual to sub-sample a mini-batch of B data points at each parameter update:

$$\theta_{n+1} = \theta_n - \eta \sum_b \nabla \log p(\boldsymbol{x}^{(b)}; \theta_n) \ . \tag{28}$$

This induces randomness in the training process since the data points are randomly sampled from the whole dataset. Therefore, the resulting algorithm is called stochastic gradient descent (SGD).

2.2 The Bayesian way

We discussed two possible strategy to set the parameters: We can either find the value of the parameters from previous literature and other sources of prior knowledge or we can learn their values from a dataset with methods such as maximum likelihood with SGD. Is there a middle way? Can we integrate prior knowledge with new information from the data? yes we can! Using Bayesian statistics. The basic idea is to "promote" the parameter θ to a proper variable in the model that follows a *prior distribution* $p(\theta)$. This gives us a joint model

$$p(\boldsymbol{x}, \theta) = p(\boldsymbol{x}; \theta) p(\theta) \ .$$

We can now use the definition of conditional probability to obtain the probability of the parameters given the data:

$$p(\theta \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x}, \theta)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}; \theta) p(\theta)}{\int p(\boldsymbol{x}; \theta) p(\theta) \mathrm{d}\theta} \ . \tag{29}$$

This is nothing less than the famous Bayes' theorem, which is a trivial mathematical consequence of the concept of conditional probability but has deep implications and offers a powerful method to estimate parameters.

2.3 Maximum-a-posteriori

Computing the full posterior distribution $p(\theta \mid \boldsymbol{x})$ can be very challenging since it requires a potentially high dimensional integral. Furthermore, sometimes we simply want a single good value for the parameter θ instead of the whole posterior distribution. A convenient possibility is to define a loss function that is minimized by the mode of the posterior. This leads to the maximum-a-posterior loss:

$$\mathcal{L}(\theta)_{\text{MAP}} = \sum_{j}^{J} \log p(\boldsymbol{x}^{(j)}; \theta) + \log p(\theta) \ . \tag{30}$$

This MAP loss can be minimized by SGD in the same way we discussed for maximum likelihood.

2.4 Priors and regularization

It is easy to see that using informed prior distributions can resource overfitting since our prior knowledge about the parameter can lead to a more meaningful interpretation of the data. However, even uninformative priors can often help since they can "filter out" extreme dependencies in the training set which are often spurious and would not generalize to data. For example, consider again the linear regression model of house prices:

$$p(y \mid \boldsymbol{x}; W, b) = \prod_{j} \mathcal{N}(y^{(j)}; W\boldsymbol{x}^{(j)} + b, \sigma^2) \ . \tag{31}$$

If the number of regressors is large compared with the number of datapoints, it is normal than some regressors will appear to be strongly correlated with the price by pure chance. This results in some very high weights in the maximum likelihood estimate of W that reduce generalization performance. We can ameliorated this problem by assigning to W a prior distribution centered at zero, so that the weights will be shrank by the prior away from the high values. For example, we can use a uncorrelated multivariate normal distribution:

$$p(W) = \mathcal{N}(W; 0, \mathrm{I}\sigma_0^2) \ . \tag{32}$$

This prior leads to the following maximum-a-posteriory loss function:

$$\mathcal{L}(W, \boldsymbol{b})_{\text{MAP}} = \frac{1}{2\sigma^2} \sum_{j}^{J} (y^{(j)} - (W\boldsymbol{x}^{(j)} + \boldsymbol{b}))^2 + \frac{1}{2\sigma_0^2} \sum_{k} W_k^2 \ . \tag{33}$$

As you can see, this loss discourage high values of the weights since there is a term that is proportional to the square of the weights.

3 Latent variables

Usually not all the variables in a probabilistic model are directly observable. The variables in a probabilistic model that are not observed directly are called *latent variables.* As an example, consider a simple random walk model of the position of a weather balloon floating in the sky:

$$p(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1}; x_t, \nu^2) \ , \tag{34}$$

where x_t denotes the position of the balloon at time t. We cannot observe the position of the balloon directly but we have a radar that produces noisy measurements. We can model the radar measurement using a simple Gaussian emission model:

$$p(y_t \mid x_t) = \mathcal{N}(y_t; x_t, \sigma^2) \ . \tag{35}$$

We can now write the joint model of balloon position and radar measurements:

$$p(\boldsymbol{y}, \boldsymbol{x}) = p(x_0)\mathcal{N}(y_0; x_0, \sigma^2) \prod_{t=1}^{T} \mathcal{N}(x_{t+1}; x_t, \nu^2)\mathcal{N}(y_t; x_t, \sigma^2) \tag{36}$$

where $p(x_0)$ is a prior distribution over the initial position. In this example the positions x_t are the latent variable while the radar measurements y_t are observables.

3.1 Inference in models with latent variables

It is very natural to take a Bayesian perspective when working with models with latent variables. The posterior of the late variables given the observation is obtained using Bayes' theorem:

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} \ . \tag{37}$$

Of course computing the exact posterior is again impossible in most real-world models. However, the MAP approach is often not satisfactory as it is often crucial to quantify uncertainty over the value of the latent variables. For example, imagine a medical diagnosis situation of a patient whose symptoms are both compatible with a serious condition A or a harmless condition B. Imagine now that the treatment for A is very invasive. The decision by the doctor to prescribe the treatment cannot only depend only on the fact that A is more likely than be but it should also depend on the relative posterior probabilities. In fact, if the uncertainty is high, the doctor will probably decide to do some other tests instead of proceeding with the invasive treatment. In the next lectures, we will introduce powerful methods to approximate the posterior in complex probabilistic models.