Multivariate Statistics – Density estimation and generative adversarial networks

Assume a random variable X in $\mathbb{R}^p$ has unknown density p_{data}. There are several ways to estimate the density $p = p_{data}$ [1] from data samples $x_1, \ldots, x_n$, the most prominent perhaps being kernel smoothing. A kernel density estimate for n observations $x_1, \ldots, x_n$ would be of the form

$$\hat{p}_{data}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big),$$

where h is the so-called bandwith and K is a kernel, that is a non-negative function $\mathbb{R}^p \mapsto \mathbb{R}$ with integral 1. An example would be a Gaussian kernel

$$K(u) = (2\pi)^{-p/2} \exp(-\|u\|_2^2/2).$$

More important than the choice of the kernel is the choice of the bandwidth. Generally, one can use cross-validation or variations to choose the optimal bandwidth. Kernel methods generally require a sample size n that grows exponentially with dimension p (depending on the precise assumptions about the density f).

Classification approach to density estimation. Assume instead we settle for the less ambitious goal of being able to sample from the true (but unknown) density p_{data} that generated the samples $x_1, \ldots, x_n$, at least approximately. Suppose we have a candidate random variable with density p_g from which we **can** sample. How can we estimate the unknown density p_{data} if we can sample from p_g? One approach is via classification. Let a distribution for (Y, U) with $U \in \mathbb{R}^p$ and $Y \in \{0, 1\}$ be defined in the following way:

$$\begin{aligned} P(Y = 0) &= P(Y = 1) = 1/2 \\ U|Y = 1 &\sim \quad p_{data} \text{ – 'real' data samples} \\ U|Y = 0 &\sim \quad p_g \text{ – 'fake' data samples} \end{aligned}$$

In other words, the indicator variable Y is sampled uniform in $\{0, 1\}$. Conditional on $Y = 0$, we sample random variable $U \sim p_g$ and conditional on $Y = 1$, we sample from $U \sim p_{data}$ (by using the observations $x_1, \ldots, x_n$).

Suppose we want to classify a sample U as to whether the underlying sample is "fake" $(Y = 0)$ or "real" $(Y = 1)$. We want to find a "discriminator" function

$$d : \mathbb{R}^p \mapsto [0, 1].$$

that returns the conditional probability of $Y = 1$, conditional on the observed sample $U = u$,

$$d(u) = P(Y = 1|U = u).$$

[1] using p_{data} here to clearly separate it from dimension p

We can approximate d with $\hat{d}$ by training a classifier (for example using logistic regression, Random Forest or a neural network) with the n samples from f and n samples from g (with corresponding n values 0 and n values 1 for the observations of Y).

Note that

$$d(u) = \frac{p_{data}(u)}{p_{data}(u) + p_g(u)}.$$

Hence

$$p_{data}(u) = p_g(u)\frac{d(u)}{1 - d(u)}.$$

We could hence in principle approximate p_{data} as

$$\hat{p}_{data}(u) = p_g(u)\frac{\hat{d}(u)}{1 - \hat{d}(u)}.$$

This way a density estimation problem is turned into a classification problem on which we can use any standard classification algorithm such as linear models, tree ensembles, deep networks.

The success of this approach will depend on how well the conditional probability d can be estimated. While kernel regression will in general suffer heavily as the dimension of the problem increases (as discussed above), an advantage of classification can be that it might still succeed in higher dimensions. For example, say the density p_{data} is really just a function of a small number $s \ll p$ of variables. Then some classification algorithms such as tree ensembles or sparse linear models work almost as well as if the relevant subset of variables were known in advance while Kernel regression would suffer the impact of the full dimension p.

Connection to TV-distance. We can turn an estimate $\hat{d}$ of d into a binary classification $\hat{Y} = \hat{Y}(u) \in \{0, 1\}$ by setting

$$\hat{Y}(u) = \begin{cases} 1 & \text{if } \hat{d}(u) \geq c \\ 0 & \text{if } \hat{d}(u) < c \end{cases}$$

For a given "proposal" density p_g, we would like to maximize the rate of correct classification

$$P(\hat{Y} = Y).$$

What is the best rate one can achieve for the densities p_{data} and p_g (with corresponding distributions P_{data} and P_g)? Let $\mathcal{H}$ be the set of all measurable functions from $\mathbb{R}^p$ to $\{0, 1\}$. Then the question is the value of

$$\max_{h \in \mathcal{H}} P(h(U) = Y),$$

where the probability is with respect to the mixture distribution for (Y, U) as mentioned above. Let A be the subset of values of u for which we set $\hat{Y} = 1$ (and $\hat{Y} = 0$ for A^c). Then

$$\begin{aligned} P(h(U) = Y) &= P(h(U) = Y|Y = 1)P(Y = 1) + P(h(U) = Y|Y = 0)P(Y = 0) \\ &= \frac{1}{2}\Big[\int_A p_{data}(u)du + \int_{A^c} p_g(u)du\Big] \\ &= \frac{1}{2}\int_A \Big[p_{data}(u) - p_g(u)\Big]du + \frac{1}{2}, \end{aligned}$$

having used that $1 = \int g(u)du = \int_A p_g(u)du + \int_{A^c} p_g(u)du$. The optimal value is achieved by setting $h(u) = 1$ iff $p_{data}(u) > p_g(u)$ and $h(u) = 0$ otherwise. This corresponds to the case $\hat{d} = d$ and $c = 1/2$ above. Hence

$$\max_{h\in\mathcal{H}} P(h(u) = Y) = \frac{1}{2}\Big[\sup_A |P_{data}(A) - P_g(A)| + 1\Big] = \frac{1}{2}\Big[TV(P_{data}, P_g) + 1\Big],$$

where $TV(P_{data}, P_g) \in [0, 1]$ is the total variation distance

$$TV(P_{data}, P_g) = \sup_A |P_{data}(A) - P_g(A)| = \frac{1}{2}\int |p_{data}(u) - p_g(u)|d$$

between the probability distributions P_{data} and P_g, with $TV(P_{data}, P_g) = 0$ if p_{data} and p_g are identical (almost everywhere).

Note that for a given "proposal" density p_g we would like to minimize misclassification error. But we would like to choose p_g to make the minimal misclassification rate as large as possible to keep the optimal d as far away from 0 and 1 as possible. In the best case $p_g = p_{data}$ and $d = 1/2$ almost everywhere. This approach is used more explicitely in generative adversarial networks.

Generative adversarial networks. The basic idea of generative adversarial networks is to generate samples from the "fake" distribution p_g as $U = g(Z)$, where $g : \mathbb{R}^q \mapsto \mathbb{R}^p$ is a parametric function (a neural network) and

$$Z \sim \Phi$$

is usually chosen to have a standard normal distribution. We will denote the density of the "fake" samples generated in this way as g_h. The function g is now ideally chosen to not allow a classifier to distinguish it from the real distribution p_{data}.

The optimisation problem is set up as

$$\text{argmin}_g \, \text{argmax}_d \, C(g, d),$$

where the objective function is given by

$$C(g, d) := E_{U\sim p_{data}}[\log d(U)] + E_{U\sim p_g}[\log(1 - d(U))],$$

and is equivalent to the likelihood of the observations[2].

The setup is "adversarial" in that d is trying to maximize the objective (by discriminating as well as possible between real and fake samples) while g is trying to minimize it (by making the distribution p_g as similar as possible to the distribution p_{data}). If g and d are parametrized by θ and w respectively, then one can follow a gradient **descent** for θ and a simultaneous gradient **ascent** for w of the objective function

$$C(g,d) = E_{U\sim p_{data}}[\log d(U)] + E_{U\sim p_g}[\log(1-d(U))].$$

For a fixed g, the optimal discriminator is of the form

$$d^*(u) = \frac{p_{data}(u)}{p_{data}(u)+p_g(u)}.$$

This can be derived by writing the objective as

$$C(g,d) = \int \Big(p_{data}(u)\log d(u) + p_g(u)\log(1-d(u))\Big)du.$$

Exchanging integration and differentiation, the optimal d^* can be seen to satisfy for (almost) all u,

$$\frac{p_{data}(u)}{d^*(u)} - \frac{p_g(u)}{1-d^*(u)} = 0,$$

which is equivalent to $d^*(u) = p_{data}(u)/(p_{data}(u)+p_g(u))$. Using the optimal discriminator we are left with

$$C(g) := \max_d C(g,d) = E_{U\sim f}\Big[\log \frac{p_{data}(u)}{p_{data}(u)+p_g(u)}\Big] + E_{U\sim p_g}\Big[\log \frac{p_g(u)}{p_{data}(u)+p_g(u)}\Big].$$

Gobal optimum is achieved if and only if $p_g = p_{data}$ (almost) everywhere. Then the realized value is $-\log 4$. Can also write

$$C(g) = -\log(4) + KL(p_{data}||\frac{p_{data}+p_g}{2}) + KL(p_g||\frac{p_{data}+p_g}{2}),$$

or, using the Jensen-Shannon divergence (JSD), as

$$C(g) = -\log(4) + JSD(p_{data}||p_g),$$

[2]using the mixture distribution above for Y,U) above, where samples from f have value $Y=1$ and samples from the "fake" p_g have value $Y=1$, the objective could also be written as

$$\text{argmin}_h \text{ argmax}_d \; E[Y\log d(U) + (1-Y)\log(1-d(u))].$$

where the Jensen-Shannon divergence is a symmetrized version of the Kullback-Leibler divergence. For general densities f and g,

$$JSD(f||g) = KL(f||\frac{f+g}{2}) + KL(g||\frac{f+g}{2}).$$

Hence optimization of the objective function leads (ideally) to a situation where $p_g = p_{data}$ (that is we sample from the right density) or (if this is impossible due to underparametrization of g) to the best possible approximation of p_{data} within the class of possible densities, where "best possible" is the density that minimizes the Jensen-Shannon divergence with the real data p_{data}.