A note on density estimation via classification

Yen-Chi Chen
University of Washington
October 3, 2025

It is known that if we have a (multivariate) density estimator, we can turn it into a regression model or use it for classification. More explicitly, suppose we want to estimate the regression function $m(x) = \mathbb{E}[Y|X = x]$ and we have a joint density estimator $\widehat{p}(x,y)$. We can then estimate m via

$$\widehat{m}(x) = \frac{\int y\widehat{p}(x,y)dy}{\int \widehat{p}(x,y)dy}$$

assuming that the evaluation of density and integration is easy.

Note that if evaluation is intractable but we are able to sample from $\widehat{p}(x,y)$, we can still approximate $\widehat{m}(x)$ via Monte Carlo methods such that we generate:

$$(\widetilde{X}_1, \widetilde{Y}_1), \cdots, (\widetilde{X}_M, \widetilde{Y}_M) \sim \widehat{p}(x,y)$$

and then apply a nonparametric regressor using these M samples. Therefore, methods for density estimation can be inverted into a regression estimator.

For classification, we consider a binary classification problem where $Z \in \{0,1\}$ is our class label and our goal is to predict the label Z with the feature X. It is well-known that the Bayes classifier under the conventional 0-1 loss is

$$\begin{aligned} c(x) &= \begin{cases} 1, & \text{if } P(Z=1|X=x) \geq P(Z=0|X=x) \\ 0, & \text{if } P(Z=1|X=x) < P(Z=0|X=x) \end{cases} \\ &= \begin{cases} 1, & \text{if } p(x|Z=1)P(Z=1) \geq p(x|Z=0)P(Z=0) \\ 0, & \text{if } p(x|Z=1)P(Z=1) < p(x|Z=0)P(Z=0) \end{cases}. \end{aligned}$$

Thus, once we have density estimators $\widehat{p}(x|Z=0), \widehat{p}(x|Z=1)$, we can convert it into a classifier

$$\widehat{c}(x) = \begin{cases} 1, & \text{if } \widehat{p}(x|Z=1)\widehat{P}(Z=1) \geq \widehat{p}(x|Z=0)\widehat{P}(Z=0) \\ 0, & \text{if } \widehat{p}(x|Z=1)\widehat{P}(Z=1) < \widehat{p}(x|Z=0)\widehat{P}(Z=0) \end{cases},$$

where $\widehat{P}(Z=z) = \frac{1}{n}\sum_{i=1}^n I(Z_i = z)$.

The above result shows that once we have a new density estimation method, we can use it for regression and classification. However, can we do the other way around that turns a density estimation problem into a classification (or a regression problem)?

It turns out that the answer is yes and here is a very simple approach to achieve it.

1 Density estimation via simulation and classification

Suppose we have $X_1, \cdots, X_n \sim p_0$ from some unknown PDF p_0 that we want to estimate and $X_i \in \mathbb{R}^d$. And we have powerful machine that can do classification pretty well (think of the modern deep neural nets). We assume that all $X_i \in [0,1]^d$ for simplicity.

We will utilize a simulation approach to turn the density estimation into a classification problem. We now simulate another sample $X_1', \cdots, X_m'$ from a known density function $q(x)$. For simplicity, we choose $q(x)$ to be the uniform distribution over the support $[0,1]^d$. Now, we combine the two samples into a new sample

$$\widetilde{X}_1, \cdots, \widetilde{X}_n, \widetilde{X}_{n+1}, \cdots, \widetilde{X}_{n+m}$$

such that

$$\widetilde{X}_i = X_i, \qquad i = 1, \cdots, n$$

and

$$\widetilde{X}_{n+i} = X_i', \qquad i = 1, \cdots, m.$$

Also, we add a 'class label' Z_i to these new observations such that $Z_i = 0$ for $i = 1, \cdots, n$ and $Z_i = 1$ for $i = n+1, \cdots, n+m$. Clearly, the label Z indicates if the observation is a simulated ($Z_i = 1$) or an actual observation ($Z_i = 0$).

Now we consider the conditional density of $\widetilde{X}$ given Z. It is clear that

$$p(\widetilde{x}|Z=0) = p_0(\widetilde{x}), \qquad p(\widetilde{x}|Z=1) = q(\widetilde{x})$$

because when $Z = 0$, our data are real data so it is from p_0 while when $Z = 1$, the data is from simulation, so it has a PDF q (which is a constant if we use the uniform distribution).

Based on $(\widetilde{X}, Z)$, we can view it as a classification problem and the odds

$$O(\widetilde{x}) = \frac{P(Z=1|\widetilde{x})}{P(Z=0|\widetilde{x})} = \frac{p(\widetilde{x}|Z=1)P(Z=1)}{p(\widetilde{x}|Z=0)P(Z=0)} = \frac{q(\widetilde{x})}{p_0(\widetilde{x})}\frac{m}{n}.$$

Thus,

$$p_0(\widetilde{x}) = \frac{q(\widetilde{x})}{O(\widetilde{x})}\frac{m}{n}$$

and if we choose $m = n$ and use q to be the PDF of uniform distribution over $[0,1]^d$, we obtain

$$p_0(\widetilde{x}) = O^{-1}(\widetilde{x}). \tag{1}$$

Mimicking the idea of logistic regression, we model the log-odds as

$$\log O(\widetilde{x}) = f_\theta(\widetilde{x})$$

and using equation (1), we obtain a model

$$p_0(\widetilde{x}) = e^{-f_\theta(\widetilde{x})}. \tag{2}$$

As a result, we can estimate p_0 via

$$\widehat{p}_0(x) = e^{-f_{\widehat{\theta}}(x)}. \tag{3}$$

Equation (3) shows how we turn a classification problem (generative classifier) into a density estimator and it also shows how the accuracy of classification influences the accuracy of density estimation. More explicitly, suppose $\widehat{\theta} \xrightarrow{P} \theta^*$ for some θ^* and let $p_*(x) = e^{-f_{\theta^*}(x)}$ be the corresponding target. Then we immediately have

$$\begin{aligned} \left| \frac{\widehat{p}_0(x) - p_*(x)}{p_*(x)} \right| &= \left| e^{-(f_{\widehat{\theta}}(x) - f_{\theta^*}(x))} - 1 \right| \\ &\leq 2|f_{\widehat{\theta}}(x) - f_{\theta^*}(x)| \end{aligned}$$

when $f_{\widehat{\theta}}(x) - f_{\theta^*}(x) \approx 0$. This provides a bound on the stochastic variation of $\widehat{p}_0(x)$. To get the rate toward $p_0(x)$, we just need to control the bias (approximation error) $\frac{|p_*(x) - p_0(x)|}{p_0(x)}$.

Estimation of θ can be done in a conventional logistic regression model procedure. Recall that $O(\widetilde{x}) = e^{f_\theta(\widetilde{x})}$. This implies that

$$P(Z = 1|\widetilde{x}) = \frac{e^{f_\theta(\widetilde{x})}}{1 + e^{f_\theta(\widetilde{x})}}, \qquad P(Z = 0|\widetilde{x}) = \frac{1}{1 + e^{f_\theta(\widetilde{x})}}$$

Therefore, the log-likelihood of $(\widetilde{x}, z)$ is

$$\ell(\theta|\widetilde{x}, z) = z f_\theta(\widetilde{x}) - \log(1 + e^{f_\theta(\widetilde{x})})$$

and our estimator is

$$\widehat{\theta} = \mathsf{argmax}_\theta \sum_{i=1}^{2n} \ell(\theta|\widetilde{X}_i, Z_i).$$

This approach is particularly useful in modern age of AI because we have many powerful tools for training a good classifier. As long as we can predict the odds/probability of the labels, we can then convert it into a density estimator.

Note that this procedure is somewhat related to the generative adversarial networks (GANs). In GANs, we adjust the sampling distribution q and consider all possible classifiers to estimate the odds. If all classifiers are performing badly, i.e., the odds is 1 everywhere, then the distribution $q = p_0$ and we obtain the true generative model.

2 Conditional density estimation

The above simulation and classification method can be applied to estimate the conditional density as well. The following paper provides a comprehensive analysis on this idea:

> CINDES: Classification induced neural density estimator and simulator. Dehao Dai, Jianqing Fan, Yihong Gu, Debarghya Mukherjee. arXiv: 2510.00367 https://www.arxiv.org/abs/2510.00367.

Here is how this idea is applied to estimating a conditional density. Suppose we have two sets of variables

$$(X_1, Y_1), \cdots, (X_n, Y_n) \sim p_0(x, y)$$

such that $(X, Y) \in \mathbb{R}^{d_x + d_y}$. Our goal is to estimate the conditional PDF $p_0(y|x)$.

The idea in [CINDES] is that we simulate Y only. Namely, we generate

$$Y_1', \cdots, Y_n' \sim q(y)$$

from some known density q such as the uniform. Then our 'simulated' data is

$$(X_1, Y_1'), \cdots, (X_n, Y_n').$$

Note that in the simulated data, X and Y' are *independent*. This independence is useful in the later derivation.

Similar to what we have done in the previous section, we combine the two dataset and add a class label:

$$(\widetilde{X}_1, \widetilde{Y}_1, Z_1), \cdots, (\widetilde{X}_{2n}, \widetilde{Y}_{2n}, Z_{2n}),$$

where $\widetilde{X}_i = X_i, \widetilde{Y}_i = Y_i, Z_i = 0$ for $i = 1, \cdots, n$ and $\widetilde{X}_i = X_{i-n}, \widetilde{Y}_i = Y'_{i-n}, Z_i = 1$ for $i = n+1, \cdots, 2n$. This leads to an interesting density of $\widetilde{X}, \widetilde{Y}$ given Z. When $Z = 0$, we have

$$p(\widetilde{x}, \widetilde{y}|Z = 0) = p_0(\widetilde{x}, \widetilde{y}) \tag{4}$$

and when $Z = 1$, the independence between simulated Y' and X implies

$$p(\widetilde{x}, \widetilde{y}|Z = 1) = p_0(\widetilde{x})q(\widetilde{y}). \tag{5}$$

Under this construction, the odds of Z versus $\widetilde{X}, \widetilde{Y}$ is

$$\begin{aligned} O(\widetilde{x}, \widetilde{y}) &= \frac{P(Z = 1|\widetilde{X} = \widetilde{x}, \widetilde{Y} = \widetilde{y})}{P(Z = 0|\widetilde{X} = \widetilde{x}, \widetilde{Y} = \widetilde{y})} \\ &= \frac{P(\widetilde{X} = \widetilde{x}, \widetilde{Y} = \widetilde{y}|Z = 1)P(Z = 1)}{P(\widetilde{X} = \widetilde{x}, \widetilde{Y} = \widetilde{y}|Z = 0)P(Z = 0)} \\ &= \frac{p_0(\widetilde{x})q(\widetilde{y})}{p_0(\widetilde{x}, \widetilde{y})} \\ &= \int_{\mathcal{Y}} dy / p_0(\widetilde{y}|\widetilde{x}) \end{aligned}$$

when q is the uniform distribution over $\mathcal{Y}$, the support of Y. Note that if Y has a support $[0, 1]^{d_y}$, the above integral $\int_{\mathcal{Y}} dy = 1$.

When Y is supported on $[0, 1]^{d_y}$, we conclude that

$$p_0(\widetilde{y}|\widetilde{x}) = O^{-1}(\widetilde{x}, \widetilde{y}) \tag{6}$$

and we can train a classifier using $\{(\widetilde{X}_i, \widetilde{Y}_i, Z_i)\}$ and convert it into a conditional density estimator. If Y is not supported on the cube, we need to adjust it by the integral $\int_{\mathcal{Y}} dy$.

3 Tukey's factorization

Here is an alternative view of this simulation and classification approach from the Tukey's factorization.

Suppose we have two sets of observations $X_1, X_2, \cdots, X_n \sim p_0$ and $X'_1, \cdots, X'_m \sim q_0$ such that they are the same variables but collected in different populations. We can then combine these observations into $\widetilde{X}_1, \cdots, \widetilde{X}_{n+m}$ such that the first n observations are just $X_1, \cdots, X_n$ and the remaining observations are $X'_1, \cdots, X'_m$ and introduce a label Z_i such that $Z_i = 0$ for $i = 1, \cdots, n$ and $Z_i = 1$for $i = n+1, \cdots, n+m$.

Our goal is to estimate p_0. Using the Bayes rule, one can easily see that

$$O(\widetilde{x}) = \frac{P(Z=1|\widetilde{x})}{P(Z=0|\widetilde{x})} = \frac{q_0(\widetilde{x})P(Z=1)}{p_0(\widetilde{x})P(Z=0)} = \frac{q_0(\widetilde{x})}{p_0(\widetilde{x})}\frac{m}{n}.$$

Therefore,

$$p_0(\widetilde{x}) = \frac{m}{n} \cdot q_0(\widetilde{x}) O^{-1}(\widetilde{x}). \tag{7}$$

Namely, we can represent the density p_0 as the density q_0 scaled by the inverse of odds. This result is known as the *Tukey's factorization*.

Note that this problem also occurs in transfer learning. In transfer learning, we often have $m >> n$ and/or have a pretty accurate estimator $\widehat{q}_0$ (think of it as coming from pre-training on a large but public dataset). So the question in transfer learning is how do we turn this accurate estimator into estimating the density of the target population p_0.

Clearly, equation (1) is the is a special case of the Tukey's factorization in equation (7) when q_0 is chosen to be a uniform distribution and $m = n$. In fact, equation (7) can be viewed as a general form of equation (1) where we are allowed to use other density function. One key requirement for q_0 is that it has to cover the support of p_0 (which implies that the odds is bounded away from 0).

4 Sampling using Langevin dynamics

Suppose we have a density estimator using equation (3): $\widehat{p}_0(x) = e^{-f_{\widehat{\theta}}(x)}$. Now we want to study the problem of sampling from $\widehat{p}_0$. It turns out that classification-based method introduced in previous sections offers a nice way for sampling via the Langevin dynamics.

Since the density estimator has a interesting exponential form, we consider the log-density:

$$\log \widehat{p}_0(x) = -f_{\widehat{\theta}}(x).$$

If we can compute the gradient of the log-density, we can use the Stochastic Langevin dynamics to sample from $\widehat{p}_0(x)$. Starting at an initial point $x^{(0)}$, the Langevin dynamics is creating a random sequence of variables $x^{(0)}, x^{(1)}, x^{(2)}, \cdots$ via

$$x^{(t+1)} = x^{(t)} + \gamma \nabla_x U(x^{(t)}) + \sqrt{2\gamma} W_t,$$

where $W_1, W_2, \cdots \sim N(0, \mathbf{I})$ are IID Gaussian noises and $U(x) > 0$ is a function guiding the dynamics. In statistical mechanics, $-U(x)$ is called the potential energy (function). When $\gamma \to 0$, the sequence $\{x^{(0)}, x^{(1)}, x^{(2)}, \cdots\}$ has a stationary distribution proportional to $e^{U(x)}$.

To use this in our scenario, we choose $U(x) = \log \widehat{p}_0(x)$, which leads to

$$x^{(t+1)} = x^{(t)} + \gamma \nabla_x \log \widehat{p}_0(x^{(t)}) + \sqrt{2\gamma} W_t.$$

And the resulting sequence will converge to $e^{\log \widehat{p}_0(x)} = \widehat{p}_0(x)$.

This idea is particularly useful in our case because

$$\nabla_x \log \widehat{p}_0(x) = -\nabla_x f_{\widehat{\theta}}(x)$$

may be easily computed. In particular, if we use the simplest logistic regression, $f_\theta(x) = \theta^T x$ so $\nabla_x f_{\widehat{\theta}}(x) = \widehat{\theta}$.