Non-Parametric Density Estimation

S. V. N. Vishwanathan

June 9, 2014

So far we have concentrated on drawing samples from a given distribution. Now we turn our attention to the inverse problem. Given $X_1, \ldots, X_n$ drawn from an unknown probability distribution we want to estimate the distribution.

Parametric methods assume that the data is drawn from some well known distribution (say Gaussian) and the task is to find the best parameter that explains this data. This is particularly useful if we have some domain knowledge about the distribution from which the data is drawn.

Unfortunately, in many situations we do not know anything about the underlying form of the distribution. In such cases we employ non-parametric methods, which do not make any assumptions about the underlying form of the distribution.

1 Histogram Methods

For simplicity we start with one dimensional random variables. Given a sample $X = \{X_1, \ldots, X_N\}$ partition it into distinct bins of width $\Delta_i$, and count the number $n_i$ of observations in each bin. To turn these counts into a probability density set

$$p_i = \frac{n_i}{N\Delta_i}.$$  

Verify for yourself that $p_i$ integrates to 1. Typically we choose $\Delta_i = \Delta$ for all bins. The key issue in an histogram estimate is that we need to choose $\Delta$ appropriately. If $\Delta$ is too small, then the distribution is very spiky and does not exhibit any local structure. On the other hand, if $\Delta$ is too large, then the distribution is smoothed out and we again do not observe any local structure.

The advantages of histogram methods are as follows:

- One can decide to discard the data and retain only the histograms.
- They can be easily adapted to estimate density in an online fashion.
- They are useful for quick visualization of data.

The main disadvantages are as follows:
• The estimated density is discontinuous at the bin edges.

• The method scales badly with dimensions. If we divide each dimension into $M$ bins, and if there are $D$ dimensions then we have to work with $M^D$ bins. This in turn implies that the amount of data needed to get a meaningful estimate is prohibitive.

Some lessons we can learn are as follows:

• To estimate the probability density at a location it is worthwhile to consider the data points which lie within some local neighborhood of that point.

• The bin width acts as a smoothing parameter. Of course, it needs to be tuned properly to achieve optimal performance (model selection problem).

2 Density Estimation Primer

Let $p(x)$ be a probability density on a $D$ dimensional space. Given samples drawn from $p(x)$ we want to estimate $p$. Let $R$ be a small region such that

$$P = \int_R p(x) \, dx.$$ 

If we obtain a data set with $N$ samples drawn from $p(x)$, the probability that $K$ of them fall in region $R$ is given by the Binomial distribution

$$\text{Bin}(K|N,P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K}.$$ 

The mean fraction of points falling inside the region is $E[K/N] = P$, while the variance is given by $\text{Var}[K/N] = P(1-P)/N$. As $N \to \infty$ the variance goes to 0 and hence the estimate peaks around the expectation. We therefore get

$$K \approx NP.$$ 

If we assume that $R$ is so small that $p(x)$ is constant over $R$, then

$$P \approx p(x) \cdot V,$$ 

where $V$ is the volume of $R$. Rearranging we obtain

$$p(x) \approx \frac{K}{NV}.$$  

Note that the above approximation depends on two contradictory assumptions. On one hand we want $R$ to be so small that $p(x)$ is constant over $R$. In other words, we want $V \to 0$. On the other hand, we want the number of points which fall in $R$ to be so large that the Binomial assumption is valid. In other words, we want $N \to \infty$. In theory, the approximation (1) depends on the rate at which $N \to \infty$ and $V \to 0$. In practice we can fix this in two different ways:
• Fix $V$ and vary $K$. This yields the kernel density estimator.

• Fix $K$ and vary $V$. This yields the $k$-nearest neighbor.

In fact, under appropriate conditions, it can be shown that both of them converge in the limit as $N \to \infty$.

3 Kernel Density Estimators

Kernel density estimators are also sometimes called Parzen window estimators. Define the unit cube centered around the origin:

$$k(u) = \begin{cases} 1 & \text{if } |u_i| \leq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

Observe that $k \left( \frac{x-x_n}{h} \right)$ is 1 if and only if $x_n$ lies inside a cube of size $h$ centered around $x$. If we let

$$K = \sum_{n=1}^{N} k \left( \frac{x-x_n}{h} \right),$$

then one can use (1) to estimate $p$ via

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{h^D} k \left( \frac{x-x_n}{h} \right),$$

where $h^D$ is the volume of the hypercube of size $h$ in $D$ dimensions. By symmetry, we can interpret this equation as the sum over $N$ cubes centered around $N$ data points $x_n$.

To prevent the estimate from being discontinuous, one can instead use a smooth kernel. A common choice is the Gaussian kernel

$$\hat{p}(x) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} \exp \left( -\frac{||x-x_n||^2}{2h^2} \right).$$

Essentially, we place a Gaussian over each data point, add up the contributions over the whole dataset, and divide by $N$ to normalize the density. There is nothing special about the Gaussian, we can use any other kernel such that $k(u) \geq 0$ and \( \int k(u) \, du = 1 \).

To estimate the error of the kernel density estimator at a single point we measure the mean square error:

$$\text{MSE}_x(\hat{p}) = \mathbb{E}[(\hat{p}(x) - p(x))^2],$$

which can be rewritten as the famous bias variance trade-off

$$\text{MSE}_x(\hat{p}) = \mathbb{E}[(\hat{p}(x) - p(x))^2] + \text{Var}[\hat{p}].$$
Note the expectation is with respect to $p(x)$. If $\int k dt = 1$, $\int tk(t) dt = 0$, $\int t^2 k(t) dt = k_2 \neq 0$, and $R(k) = \int k^2(t) dt$ then

$$\text{Bias}(\hat{p}) = \frac{1}{2} h^2 p''(x) k_2 + O(h^4)$$
$$\text{Var}(\hat{p}) = \frac{p(x) R(k)}{nh} - \frac{p(x)^2}{n} + O(h/n).$$

Therefore, for consistency (bias and variance go to zero) we need to set $h \to 0$ and $nh \to \infty$.

4 \  \textit{K-Nearest Neighbors}

One problem of kernel density estimation is that the kernel width is fixed for all kernels. This means that in regions of high density we might over-smooth things, while missing some local structure in the low density regions.

In $K$-nearest neighbor estimation, instead of fixing $V$ as we did before, we fix $K$ and find the appropriate radius of a ball centered around a data point, such that $K$ samples lie within this ball. Here $K$ is a tunable parameter, with higher values leading to smoother estimates and lower values giving us spiked estimates. One thing to bear in mind is that the $K$-nearest neighbor method is, strictly speaking, not a density estimation method. It does not give us a normalized density.

4.1 \  \textit{K-Nearest Neighbor for Classification}

Assume that we are given a set of $N$ points, and suppose $N_k$ of these points are from class $C_k$. In other words, $N = \sum_k N_k$. To classify a given point $X$ we draw a sphere around $X$ which contains exactly $K$ points. Let $V$ denote the volume of the sphere and let it contain $K_k$ points from class $C_k$. Then

$$P(X|C_k) = \frac{K_k}{N_k V}.$$ 

Similarly, $P(X) = \frac{K}{N V}$, while the prior $P(C_K) = \frac{N_k}{N}$. Using Bayes rule

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} = \frac{K_k N_k}{N V} = \frac{K_k}{K}.$$ 

Therefore, to classify a new point we simply look at the class membership of the $K$ neighbors and assign the point to the class which occurs most frequently (break ties arbitrarily). If $K = 1$ we obtain the nearest neighbor rule, which assigns a point to the same class as its nearest neighbor.

One problem with $K$ nearest neighbor is that the entire dataset needs to be stored. To classify every point we need to perform $O(N)$ work.