Kernel density estimation and its applications

Chiwoo Park¹

¹Advanced Metrology Lab., Texas A&M University

Chiwoo’s Lecture Series on Kernel Methods, 2008
1 Introduction
   - A distribution is essential
   - What if we don’t know distributions

2 Kernel density estimation (KDE)
   - Concept of Kernel Density Estimation
   - Choosing a bandwidth
   - R Example

3 SVM and KDE
   - Relationship
Some important questions

■ What is the motivation of Kernel Density Estimation?
■ What is the basic idea of Kernel Density Estimation?
■ Which drawbacks of a histogram can be relieved by Kernel Density Estimation? Which ones cannot?
■ How can we choose the bandwidth?
■ Can you write a R-code for Kernel Density Estimation?
■ Is there any relationship between Kernel Density Estimation and SVM?
No population parameter, no hypothesis

Definition
A hypothesis is a statement about a population parameter.

Example
Let $X_1, \ldots, X_n$ be a random sample from a $n(\theta, 1)$ population.

\[ H_0 : \theta = \theta_0 \text{ versus } H_1 : \theta \neq \theta_0 \] (1)

Example
Let $X_1, \ldots, X_n$ be a random sample from a density $f$. What is the mean of population?

\[ H_0 : \mu = \mu_0 \text{ versus } H_1 : \mu \neq \mu_0 \] (2)
No population parameter, no hypothesis

What is the probability that $H_0$ is true?

For the previous examples,

- Test Statistic: $\overline{X} = \frac{1}{n} \sum X_i$

- What is the distribution of $\overline{X}$?
  It depends on the distribution of a random variable $X$. 
Posterior? Prior?

Bayesian classifier

Let $x$ be a sample point. The bayesian classifier is given by:

$$P(y = i | x) = \frac{P(x | y = i)P(y = i)}{P(x)}$$  \hspace{1cm} (3)

How can we get the posterior distribution $P(x | y = i)$ and the prior distribution $P(y = i)$ from a random sample $(X_1, Y_1), \ldots, (X_n, Y_n)$?

Example

We are given a training data of $n$ sample points $(x_1, y_1), \ldots, (x_n, y_n)$. Now, we have a new point $x_{new}$. What is the most credible class for given $x_{new}$?
Assume the distribution

Parametric approach

1. Assume the distribution is a normal distribution or something else.
   - normal, chi-square, F-distribution, t-distribution
   - Gaussian mixture

2. Estimate the parameters of the distribution assumed from a random sample.

Example
We’re given a random sample of the points $X_1, \ldots, X_n$. We assume it’s from $n(\theta, 1)$. The best unbiased estimator of $\theta$ is $\overline{X}$. 
Assume the distribution

But, the problem is..

- Common parametric forms do not always fit the densities actually encountered in practice.
- In addition, most of the classical parametric densities are unimodal, whereas many practical problems involve multi-modal densities.
- Non-parametric methods can be used with arbitrary distributions and without the assumption that the forms of the underlying densities are known.
- When we use a mixture, how many components are necessary? (the number of modes)
Non-parametric approach

Estimate a whole distribution

- Parameter Estimation: a point estimation ⇔ Density estimation: a function estimation
- No parameter, so called non-parametric density estimation
- No assumption on the shape of the distribution
Non-parametric approach

Suppose \( \{X_1, \ldots, X_n\} \) is a random sample from a probability density function \( f(x) \). Let \( A \) be a small region where \( f(x) \) does not vary significantly. Then, the density at \( x \in A \) can be approximated by:

\[
P(A) = \int_A f(t)dt \approx f(x)\lambda(A) \rightarrow f(x) \approx \frac{\int_A f(t)dt}{\lambda(A)} \quad (4)
\]

Suppose that the \( k \) points among \( n \) sample points are in \( A \). Then, this probability is

\[
\binom{n}{k} P(A)^k (1 - P(A))^{n-k} \quad (5)
\]

Here, the MLE of \( P(A) \) from the pmf is \( \frac{k}{n} \).

So, the density at \( x \in A \) can be approximated by:

\[
f(x) \approx \frac{\int_A f(t)dt}{\lambda(A)} = \frac{k}{n\lambda(A)} \quad (6)
\]
Histogram

How to choose $A$

- Histogram: Partition the space into a number of equally-sized cells (bins).

\[
f(x) = \frac{1}{n} \text{ the number of the points in the bin containing } x \quad \text{volume of a bin}
\]

* These figures are from http://www.maths.uwa.edu.au/duongt/seminars/intro2kde/.

Chiwoo Park, Advanced Metrology Lab.

June 27th 2008
Histogram

But, histograms

- are not smooth (step function)
- depend on end points of bins
- depend on width of bins
k-Nearest Neighborhood

How to choose $A$

- k-Nearest Neighborhood (k-NN): Choose the smallest hypersphere $A$ so that it encloses a total of $k$ sample points.

\[ f(x) = \frac{1}{n} \frac{k}{\frac{\pi^{d/2}}{\Gamma(d/2+1)}} R^d \]

- $R$: radius of $A$ centered at $x$
- $d$: the dimension of space

* This figure is made by Ricardo Gutierrez-Osuna, Wright State University
k-Nearest Neighborhood

kNN is not satisfactory because..

- the estimates are prone to local noise
- the method produces estimates with very heavy tails
- Since $R$ is not differentiable, the density estimate might have discontinuities
- the resulting density is not a true probability density since its integral over all the sample space diverges
Kernel Density Estimation

- Motivation: relieve the drawbacks of histograms.
  - No end points of bins: Bins are generated dynamically with a center of each data point.
Kernel Density Estimation

- Motivation: relieve the drawbacks of histograms.
- Smooth and Continuous: Use a continuous function for building blocks for bins.

Box-kernel estimate

Smooth-kernel estimate
Kernel Density Estimation

Smooth function
Use a smooth kernel function $K(x)$ which satisfies the condition

$$\int K(x)dx = 1$$

(7)

Usually, but not not always, $K(x)$ will be a radially symmetric, unimodal probability density function.

Example
Gaussian distribution

$$K(x) = \frac{1}{(2\pi)^{d/2}} \exp\{-\frac{1}{2}x^T x\}$$

(8)
Kernel Density Estimation

Kernel density estimate

Let \( x_1, \ldots, x_n \in \mathbb{R}^d \) be sample points from an unknown density \( f \). Then, its kernel estimate \( \hat{f} \) is:

\[
\hat{f}(x; h) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)
\]

- \( h \): a bandwidth
- Kernel function:
  - Bandwidth \( \Leftrightarrow \) Histogram:
  - Size of bins

* This figure is made by Ricardo Gutierrez-Osuna, Wright State University

Chiwoo Park, Advanced Metrology Lab. June 27th 2008
Choosing a good bandwidth is crucial in density estimation.  

- A large bandwidth: smooth ⇔ A small bandwidth: spiky

* This figure is made by Ricardo Gutierrez-Osuna, Wright State University
Minimize MSE

- We would like to find a bandwidth that minimizes the error between the estimated density and the true density.

\[
MSE(\hat{f}) = E[(f(x) - \hat{f}(x; h))^2] = (E[f(x)] - E[\hat{f}(x; h)])^2 + \text{var}(\hat{f})
\]

- Bias-Variance dilemma
  A large bandwidth: high bias ⇔ A small bandwidth: high variance

* This figure is made by Ricardo Gutierrez-Osuna, Wright State University
Kernel density estimation (KDE)

Choosing a bandwidth

Likelihood Crossvalidation

Let $h \geq 0$ be a bandwidth. For a gaussian kernel, its log likelihood is

$$\log L(h; x_1, \ldots, x_n) = - \log(n) - d \log(h) - n \log((2\pi)^{d/2})$$

$$- \frac{1}{2h^2d} \sum_{i=1}^{n} (x - x_i)^T(x - x_i)$$

$\hat{h} = 0$. (a density estimate with delta functions at each training data point.)

Use a Pseudo Likelihood Function

$$L_{pseudo}(h; x_1, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} \log g(x_i),$$

$$g(x_i) = \frac{1}{(n-1)h} \sum_{i=1, i\neq i}^{n} K\left(\frac{x_i - x_j}{h}\right)$$
Mimimize Integral of square error (MISE)

- Find the value of the bandwidth that minimizes the integral of the square error (MISE).

\[
MISE(\hat{f}) = E[\int (f(x) - \hat{f}(x; h))^2 dx]
\] (9)

- If we assume that the true distribution is a Gaussian density and we use a Gaussian kernel, it can be shown that the optimal value of the bandwidth becomes [Silverman]

\[
h_{opt} = 1.05sn^{-1.5},
\] (10)

where \(s\) is the sample standard-deviation.

- Rule of Thumb for more robustness [Silverman]

\[
h_{opt} = 0.9An^{-1.5}, A = \min\{s, IQR/1.34\}
\] (11)

* Silverman [1986], "Density estimation for statistics and data analysis".
Related package and its syntax (univariate)

stat::density(..)

- x: training data
- bw: bandwidth selector, "nrd0" (Rule-of-Thumb), "nrd" (a variation of nrd0), "ucv" (unbiased cross validation), "bcv" (cross validation), ...
- adjust: the bandwidth used is actually adjust*bw
- kernel: "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine"
- weight: numeric vector of non-negative observation weights
- width: manually setting the bandwidth
- n: the number of equally spaced points at which the density is to be estimated
- from, to: the left and right-most points of the grid at which the density is to be estimated

Example

```r
x <- seq(-4, 4, 0.01); y <- dnorm(x, mean=0, sd=1);
ln <- rnorm(1000, mean=0, sd=1);
plot(density(x=ln, bw="nrd0", kernel="gaussian", n=1000), col=5);
lines(x, y);
for(w in 1:3)
  lines(density(x=ln, width=w, kernel="gaussian", n=1000), col=w+1);
```

Chiwoo Park, Advanced Metrology Lab. June 27th 2008
Results

density.default(x = rn, bw = "nrd0", kernel = "gaussian", n = 1000)
Related package and its syntax (multivariate)

\[ \text{ks::kde(\ldots)} \]

- \( x \): training data
- \( H \): bandwidth matrix
- \( h \): scalar bandwidth (1-d only)
- \( \text{eval.points} \): points at which density estimate is evaluated

Example

```r
library(ks);
data(unicef);
H.scv <- Hscv(unicef);
fhat <- kde(unicef, H=H.scv);
plot(fhat);
plot(fhat, drawpoints=TRUE, drawlabels=FALSE, col=3, lwd=2);
plot(fhat, display="persp");
plot(fhat, display="image", col=rev(heat.colors(100)));
plot(fhat, display="filled");
```
Results
Two-class classification

We have a training dataset of

- class 0($y_i = 0$): $(x_1, y_1), \ldots, (x_n, y_n)$ and
- class 1($y_i = 1$): $(x_{n+1}, y_{n+1}), \ldots, (x_{n+m}, y_{n+m})$.

Then, we can estimate a posterior density of each class as:

- class 0($y_i = 0$)

\[
 f(x; y = 0) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)
\]

- class 1($y_i = 1$)

\[
 f(x; y = 1) = \frac{1}{mh^d} \sum_{i=n+1}^{n+m} K\left(\frac{x - x_i}{h}\right)
\]
Two-class classification

Suppose that the two priors are same as \( f(y = 0) = f(y = 1) \). Then, the decision rule for a new point \( x \) is
- if \( f(x; y = 0) \geq f(x; y = 1) \), \( x \) is from class 0.
- otherwise, it is from class 1.

The decision boundary is

\[
\begin{align*}
f(x) &= f(x; y = 0) - f(x; y = 1) \\
&= \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) - \frac{1}{mh^d} \sum_{i=n+1}^{n+m} K\left(\frac{x - x_i}{h}\right) \\
&= \sum_{i=1}^{n} \frac{1}{nh^d} y_i K\left(\frac{x - x_i}{h}\right) + \sum_{i=n+1}^{n+m} \frac{1}{mh^d} y_i K\left(\frac{x - x_i}{h}\right) \\
&= 0
\end{align*}
\]
Two-class classification

Define $\alpha_i$ by:

$$
\alpha_i = \begin{cases} 
\frac{1}{nh^d}, & i = 1, \ldots, n \\
\frac{1}{mh^d}, & i = n + 1, \ldots, n + m
\end{cases}
$$

And, let $K(x_i, x) = K\left(\frac{x - x_i}{h}\right)$.

Then, the boundary can be expressed as:

$$
f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + \sum_{i=n+1}^{n+m} \alpha_i y_i K(x_i, x)
$$

$$
= \sum_{i=1}^{n+m} \alpha_i y_i K(x_i, x) = 0,
$$

where $\sum_i \alpha_i y_i = 0$. 

Chiwoo Park, Advanced Metrology Lab. 

June 27th 2008
Bayes classifier by KDE v.s. SVM Classifier

Case: Gaussian Kernel
The following table is the comparison:

<table>
<thead>
<tr>
<th>KDE</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ \sum_{i=1}^{n+m} \alpha_i y_i K\left(\frac{x-x_i}{h}\right) = 0 ] [ x_i \Rightarrow K\left(\frac{x-x_i}{h}\right) ] [ K\left(\frac{x-x_i}{h}\right) = \frac{1}{(2\pi)^{d/2}h^d} \exp\left{ -\frac{1}{2}</td>
<td></td>
</tr>
</tbody>
</table>

- The two boundaries are exactly same except for the threshold constants.
- Do you think this match is occurred by accident?
Summary

- What is the motivation of Kernel Density Estimation?
- What is the basic idea of Kernel Density Estimation?
- Which drawbacks of a histogram can be relieved by Kernel Density Estimation? Which ones cannot?
- How can we choose the bandwidth?
- Can you write a R-code for Kernel Density Estimation?
- Is there any relationship between Kernel Density Estimation and SVM?