ICS 663: Homework 2 - Kernel Density Estimation

Christopher Mullins

October 5, 2011

Contents

1 Introduction

Kernel density estimation is a technique for deriving a non-parametric estimate for the probability density function (PDF) of a dataset. This means it does not have any assumptions about the underlying distribution of the data it is applied to. It works by considering hypercubic regions within the sample space centered around each point in the training data. These regions are determined by a parameter h that specifies the length of an edge along the hypercube.

The estimate, $\hat{p}(\mathbf{x})$, considers the distance of each point in the training sample from $\mathbf{x}$ and contributes some value to a mean. The actual calculation is:

$$\hat{p}(\mathbf{x}) = \frac{1}{nh^k} \sum_i^n \varphi \left(\frac{\mathbf{x} - \mathbf{x}_i}{h} \right).$$

where n is the number of elements in the training set, $\{\mathbf{x}_i\}$ is the training set, and $\varphi(\mathbf{u})$ is a *kernel function* that gives some indication how likely it is that $\mathbf{x}$ is in the dataset $\hat{p}(\mathbf{x})$ describes for each $\mathbf{x}_i$ in the training set. The most naive choice for φ yields 1 when $\mathbf{x}$ is in the hypercube of length h centered at $\mathbf{x}_i$ and 0 otherwise. Mathematically, this is:

$$\varphi(\mathbf{u}) = \left\{ \begin{array}{ll} 1 \quad : |u_i| < \frac{1}{2} & \forall 1 \leq i \leq n \\ 0 & \text{otherwise} \end{array} \right\}.$$

Notice that $\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) = 1$ if x is within the hypercube of length h centered at $\mathbf{x}_i$, and 0 otherwise.

This is very rarely a good choice for φ, however. Its discontinuousness can cause problems, and it's probably not often reasonable to consider a point further away as good of a match as a point that's close. A common alternative is a 0-mean gaussian:

$$\varphi(\mathbf{u}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2} \left(\mathbf{x}^T \Sigma^{-1} \mathbf{x} \right) \right).$$

Here, k is the number of dimensions in $\mathbf{x}$, and Σ is $k \times k$ matrix that is (potentially) related to the variance of the training data. Obvious choices for Σ are:

1. I_k (i.e., unit-variance)

2. $\sigma^2 I_k$, the mean of the variances for each dimension of data
3. diag $\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\right)$, a diagonal matrix a_{ij} where $a_{ii} = \sigma_i^2$ is the variance of the ith dimension.
4. cov $(\{x_i\})$

In this assignment, multivariate gaussian with Σ following case number 3 is used.

2 Implementation

My implementation is again in R. There are three source files: `common.R`, `parzen_window_estimation.R`, and `hw2.R`. `common.R` includes some utility methods that I foresee using beyond this assignment. `parzen_window_estimation.R` includes methods for creating PDF estimations for a provided dataset. `hw2.R` is a script that seamlessly runs all of the requested tasks for this assignment.

2.1 Running

It should be as simple as running the `hw2.R` script. To run from the commandline, one could use the command `R --no-save --slave < hw2.R`. The output should be the table indicating error rates.

3 Results

Results for fixed values of h are shown below. The same measurement is taken 15 times to ensure a smooth result.

Run	**Width** = 0.01	**Width** = 0.5	**Widith** = 10
1	0.509804	0.019608	0.666667
2	0.450980	0.019608	0.666667
3	0.450980	0.058824	0.666667
4	0.470588	0.039216	0.666667
5	0.392157	0.039216	0.666667
6	0.431373	0.039216	0.666667
7	0.372549	0.000000	0.666667
8	0.549020	0.039216	0.666667
9	0.431373	0.019608	0.666667
10	0.529412	0.000000	0.666667
11	0.411765	0.019608	0.666667
12	0.431373	0.058824	0.666667
13	0.333333	0.019608	0.666667
14	0.431373	0.019608	0.666667
15	0.411765	0.019608	0.666667
Mean	0.440523	0.027451	0.666667
Variance	0.057342	0.017848	0.000000

These results highlight the fact that the choice of h is very important to how successful the kernel density estimation technique is. If a window is too small, data that are even slightly distant from the training data will have near-0 posterior probability. Too large, and there will be hardly any difference between points that are very far apart.