Parzen and Knn

Monday, May 15, 2017 12:30 PM

Logistics

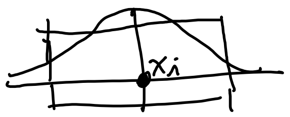
- You can work on this Lab-homework in groups of two or on your own. No teams of more than 3 students allowed.
- All the code is to be written in Matlab or Octave
- Step by step instructions are provided throughout the description of this assignment. Some code is included for your convenience. Additional action items or requirements you need to address are noted in **bold** font.
- Turn in report and code all in a zip file
  - Report should be a single document in docx or pdf format
  - Turn in all your code as .m files
- *If *any* copy-paste of these instructions/descriptions are found in your report, you will receive a penalty of 50%: your report must be a document, not a "questionnaire print-out" or "fill-in the blanks". Write your own words, be professional and creative.*

References

- Textbook Ch4
- http://www.statsoft.com/textbook/k-nearest-neighbors

1. Parzen Windows

Implement Parzen window classifier using a Gaussian Kernel



Implement Parzen Window Method

Write code for Equations 26 and 27 on page 168 of the textbook.

$$\varphi(u) = \frac{1}{\sqrt{2\pi}}\, e^{-u^2/2}$$

Eq. 26

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_n}\varphi\left(\frac{x - x_i}{h_n}\right)$$

Eq. 27

Name your function `parzen_window_gaussian`

Test it with the following:

- Using these three different values of h, i.e., h=1, h=0.6, h=0.15
- Use various sample sizes, n=1, n=10, n=100, n=1e4

  - Suggestion: You can put these values in vectors and then iterate over them, this helps automate your testing

```
hs = [1 0.6 0.15]; % this parameter controls the window width h_n
n = [1 10 100 1e4]; % total number of samples
```

- Vary the number of dimensions that you use, start with d=1 (e.g., using one feature)

```
d = 1;  % number of dimensions
```

- Use the following artificial "data sets" for starters, after you test your function with this dataset you will have to test it on the iris data set
  - Use the following vector representing data points to be tested (at which density will be estimated)

```
x = [-2:0.1:2]; % this is your vector of observations x
len_x = length(x);
```

  - Use the following to generate a random sample of size `n` with a Normal distribution

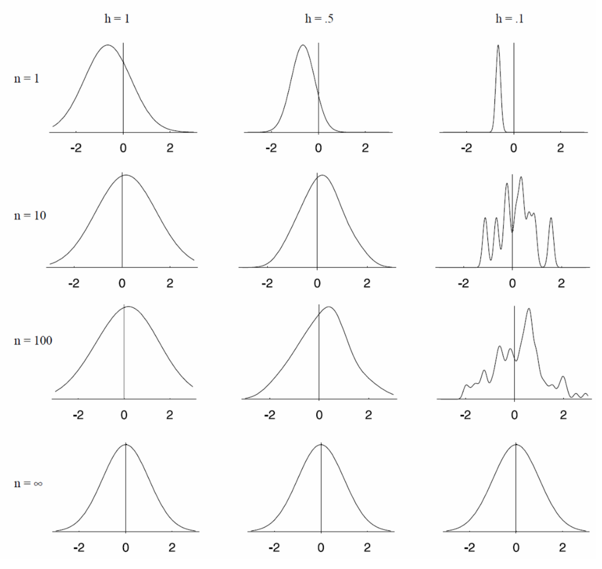
```
samples = random('normal',0,1,n,1);
```

To obtain the resulting sum (equation 27), i.e., the probability estimate for each **x**, you will compare against each of the values of samples (for each **x** iterate over all the `samples` contents).

Hint: iterate over **x**, and for each **x** compute the sum and save each corresponding estimate in a vector you can then use to generate your plots, for instance:
`p_estimate(i)=sum/n` where `i` is the variable that iterates over **x**

Plots

Generate Normal pdf plots (the should be similar to those in page 169 of your textbook -- see figure below). You can easily do this by plotting **x** vs. the contents of your `p_estimate`



Pattern Classification, Duda, Hart, 2nd E d. Figure 4.5: Parzen-window estimates of a univariate normal density using dif ferent window widths and numbers of samples.

Implement Parzen Window Discriminant

Use your `parzen_window_gaussian` to implement a classifier (discriminant) function and use it on the `Iris` dataset.
Note: several teams are having issues plotting multiple dimensions. If you are having this problem, just choose one or two dimensions at a time.

2. Knn

In this part of the lab, w e will use Knn approach to classify based on the kno wn or *training* data.

To fully under stand the beha vious of Knn classifier s, you are encouraged to implemen t the algorithm provided in the slides ma terial. Ho wever, for the purpose of this lab, it is enough t o use the Ma tlab provided implemen tations for Knn classific ation.

Use either `fitcknn` or `knnsearch` (provided with Ma tlab) and the `iris` dataset to classify three new flower observations.

For starters, follow Matlab's scrip ts provided in the documen tation of these functions.

For example and visualiz ation purposes, w e'll use sepal wit dth and leng th:

Load iris, and put the da ta `X` and the classes in `species` (this is basic ally your class t ags)

```
load iris;
```

Plot sepal da ta:

```
x_sepal = X(:, 3:4);
gscatter(x_sepal(:,1), x_sepal(:,2), species);
legend('Location','best');
```

Define and plot the ne w observation as a * mark er in the existing plot:

```
new_observation = [5 1.45];
line(new_observation(1),new_observation(2),'marker','*','color','k',.
..
   'markersize',9,'linewidth',2);
```

Use knn t o find 10 near est neighbor s:

```
[n, d] = knnsearch(x_sepal, new_observation, 'k', 10, 'distance',
'euclidean');
```

Nicely plot the f ound neighbor s:

```
line(x_sepal(n,1), x_sepal(n,2),'color',[.5 .5 .5],'marker','o',...
    'linestyle','none','markersize',10);

% these are the neighbors
x_sepal(n, :)
```

Print out nicely wha t class (category) they belong t o. Observe the r esults giv en by tabulate, the higher per centage indicates that this new point can be classified as v ersicolor

```
species(n);
tabulate(species(n))
```

Enhance our plot by drawing a circle around the neighbors:

```
ctr = new_observation - d(end);
diameter = 2*d(end);
% Draw a circle around the 10 nearest neighbors.
h = rectangle('position',[ctr,diameter,diameter],...
   'curvature',[1 1]);
h.LineStyle = ':';
hold off;
```

Use this example to implement a more general form of the classifier.

Using the approach in Homework 2 to select a random sample from the original data, separate the dataset into *training* and testing *samples*. Select **3** random observations from each of the iris categories, these will be the *testing* data or "new observations".

Repeat the process above to classify each of the randomly selected "new observations" using two features a time:

- Petal metrics
- Sepal metrics

Discuss and interpret all the plots and results you obtain. Clearly indicate what class each of the points was identified as and discuss whether they were correctly classified by comparing the results of tabulate with the original class tag.

Note:
Provide insightful discussions, focus on the WHYs not the what's. If you continue to just state the obvious (e.g., "the red points are to the left and the blue points to the right") your report will receive significant point deductions.

Rubric

Criteria	Points	Notes
Code	2p - 0p	Code should be well commented and structured
Analysis	Outstanding: 8pt Sufficient: 6pt Poor: 4pt Missing: 0pt	Include and discuss statistics and plots. Discuss your results, putting them in context of the results (plots and statistics) Focus on the WHYs not just the What's. Sloppy plots, discussions and formatting will be heavily penalized. Remember, your interpretation of the results is what matters the most! Points will be deducted for lousy redaction, spelling/grammar mistakes, typos -- professionalism is expected.