UN3106 Applied Data Mining
Spring 2019

Vincent Dorie
vjd2106@columbia.edu
Pratyay Datta
pd2511@columbia.edu

Homework 1[1]

Due: February 28, 2018

Homework submission: Homework will be collected *at the beginning of class* on the due date. Prior to that, you can leave the homework in Pratyay Datta's postbox in the Statistics Department mailroom on the 10th floor of the School of Social Work.

No late homeworks will be accepted.

Problem 1: Naive Bayes

Consider a classification problem with training data $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \ldots, (\tilde{\boldsymbol{x}}_1, \tilde{y}_1)$ and three classes $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. The sample space is $\mathbb{R}^5$, so each data point is of the form $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(5)})$. Suppose we have reason to believe that the distribution of each class is reasonably well-approximated by a spherical, unit variance Gaussian, i.e. the class-conditional distributions are $g(\boldsymbol{x} \mid \mu_k, \mathbb{I}_5)$ for class $k \in \{1, 2, 3\}$.

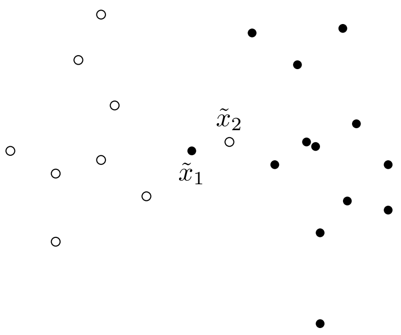
Note: If a d-dimensional random vector $(X_1, \ldots, X_d)$ has a spherical Gaussian distribution, the scalar random variables $X_1, \ldots, X_d$ are stochastically independent from each other.

1. How is the Gaussian assumption translated into a naive Bayes classifier? Write out the full formula for the estimated class label $\hat{y}_{\text{new}} = f(\boldsymbol{x}_{\text{new}})$ for a newly observed data point $\boldsymbol{x}_{\text{new}}$.
   Hint: This equation should not contain the training data, only parameters estimated from the training data.
2. How do you estimate the parameters of the model? Give the estimation equations for (a) the parameters of the class-conditional distributions and (b) the class prior $P(Y = k)$ for each class $\mathcal{C}_k$.
3. If our assumptions on the data source as described are accurate, do you expect the naive Bayes classifier to perform well? Please explain your answer.

Problem 2: Perceptron

Consider the following training data for a two-class problem in which classes are labeled +1 (open circles in the figure) and −1 (solid circles). There are $n = 22$ points in total. Clearly, these data are not linearly separable.

[1] HW thanks to Peter Orbanz.



1. What is the minimal empirical risk a linear classifier can achieve on this data set under $0-1$ loss?
2. Consider the linear classifier in $\mathbb{R}^2$ given by $(\boldsymbol{v}_{\mathrm{H}}, c)$ and two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Suppose that

$$\boldsymbol{v}_{\mathrm{H}} := \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad c := \frac{1}{2\sqrt{2}} \qquad \boldsymbol{x}_1 := \begin{pmatrix} -3 \\ 0 \end{pmatrix} \qquad \boldsymbol{x}_2 := \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

   Compute the classification result for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.
3. Suppose you run the perceptron learning algorithm (say with a constant learning rate $\alpha(n) = 1$) on the data. How is the algorithm going to behave: does it return a solution? If so, what can you say about the solution? If it does not find a solution, why not?

Problem 3: Gradient descent

In this problem, we use gradient descent to approximate the minima of the function,

$$f(x) := -\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) + \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-1}{0.5}\right)^2\right) - \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-2}{0.4}\right)^2\right)$$

1. In R, plot the function $f(x)$ for $x \in (-2, 4)$. Hint: f can be described as:

```
f <- function(x)
  -dnorm(x, 0, 1) + 0.5 * dnorm(x, 1, 0.5) - 0.4 * dnorm(x, 2, 0.4)
```

We will approximate the derivative $f'(x)$ by finite differences: we choose some small value $\delta > 0$ and compute

$$\hat{f}'_\delta(x) := \frac{f(x+\delta) - f(x-\delta)}{|(x-\delta, x+\delta)|} = \frac{f(x+\delta) - f(x-\delta)}{2\delta}$$

and use $\hat{f}'_\delta(x)$ as our estimate of $f'(x)$. The function `f.prime()` we ask you to write below implements $\hat{f}'_\delta$.

2. Write a function in R, `f.prime(x)`, to calculate the numerical derivative given the location `x`. Take the approximation window to be $\delta = 0.001$. What is the output of `f.prime(-2)`?
3. Write a function in R, `grad.des(x1)`, to peform gradient descent from the starting point `x1`. Take the step sizes $\alpha_n = 1/n$ and precision $\epsilon = 0.05$. That is, compute

$$x_{n+1} := x_n - \frac{1}{n}\hat{f}'_\delta(x_n).$$

Your function should output a list including the number of iterations N, the minima $x^* := x_N$, the minimum value $f_{\min} := f(x^*)$ and the vector of your search trajectory $(x_1, x_2, \ldots, x_N)$.

4. Start from $x_1 = -2$, what is the minimum your function finds? Plot $x_1, x_2, \ldots, x_{10}$ with the curve of f from part 1.

As we have discussed in class, a gradient descent algorithm generally finds local minima and these minima may not be global. One way to address this is to randomly start gradient descent at several different points and then compare the function values at the resulting local minima.

5. Generate 100 random starting points between $(-2, 4)$, calculate the gradient descent result for each of them. Plot the histogram of the 100 minima you found. What is the global minimum?