Up and Running with Python

Nandan Rao
2020

What is Python?

Python refers to both:

1. A program (it runs on a computer).
2. A programming language.

The program's purpose is to **interpret** the language and convert it into instructions for a computer.

Social Sciences meets ML?

> “ When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need, but not a more general one. ”

--Vladimir Vapnik

Social Sciences meets ML?

Statistical Modelling: The Two Cultures

https://projecteuclid.org/download/pdf_1/euclid.ss/1009213726

What is Python?

The program's purpose is to **interpret** the language and convert it into instructions for a computer:

1. A program (it runs on a computer).
2. A programming language.

```
> which python

> which python3
```

Classifiers for Text Mining

Now we'll go over some basic classifiers.

Outline

- Emprical risk minimization and $p(y, x)$
- $p(x)$ when X is language
- Generative vs discriminative classifiers
- Hyperparameters in BOW framework
- Towards supervising the embedded space

Emprical Risk Minimization

Consider an input space X, a discrete output space Y and a function, $g : X \to Y$ which predicts a value for y given an input x. Consider also a *loss function* $\ell : Y, Y \to \mathbb{R}$.

The *risk* of the classifier g is defined as:

$$R(g) = \mathbb{E}_{X,Y}[\ell(g(x), y)]$$

$$R(g) = \sum_{y \in Y} \int_X \ell(g(x), y)\ p(x, y)\ dx$$

Emprical Risk Minimization

We estimate the risk with its finite-sample approximation, the *empirical risk*:

$$R(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(g(x_i), y_i)$$

Note, that this is only an approximation of $\mathbb{E}_{X,Y}$ if all the pairs:

$$x_i, y_i \in p(X, Y)$$

Emprical Risk Minimization

$$x_i, y_i \in p(X, Y)$$

Implies that $p(Y|X)$ and $p(X)$ are the same in our validation distributions as they are in our target distribution.

Ways to deal with changes in any of the component parts of the joint distribution are covered in the literature of *domain adaptation*.

Naive Bayes

Let's see how a multinomial naive Bayes can be related to logistic regression:

Where π_i denotes the log probability parameters of the multinomial distribution representing $p(X|y_i)$

$$p(y=1|x) = \frac{p(x, y=1)}{\sum_y p(x, y=y_i)}$$

$$p(y=1|x) = \sigma\left(\log \frac{p(x, y_1)}{p(x, y_0)} \right)$$

$$p(y=1|x) = \sigma\left((\pi_1 - \pi_0)^T x + \log \frac{p(y_1)}{p(y_0)} \right)$$

Filter

Filter is used to remove certain elements from a list.

We can filter a list by adding an **if statement** to a list comprehension.

Like all if statements, the `if` keyword is followed by a boolean.

The element is included in the new list only if the boolean is true.

```
names = ['Foo', 'Bar', 'Baz']

def we_like(name):
    return name != 'Bar'

[n for n in names if we_like(n)]
```

Logistic Regression

Logistic regression models the posterior probability as:

$$p(y|x;\beta) = \sigma(\beta^T x)$$

where the logistic function σ is given by:

$$\sigma(n) = \frac{1}{1 + \exp^{-n}}$$

This can be fit via maximum likelihood ($\ell := -p(y|x;\beta)$) or by any other convex surragote of the 0-1 error ($\ell := 1\{g(x) \neq y\}$)

Naive Bayes

As mentioned, Naive Bayes predicts based on the posterior calculated from the modelled joint distribution:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

It's called "naive" because we make the (extreme) simplifying assumption that all the $p(x_i|y)$ are independent and thus $p(x|y) = \prod_i p(x_i|y)$.

Running Python

Python does not have a GUI, so we run it from a terminal.

Let's make sure we can find the program (the "executable").

`which` is a program, available in your terminal, that tells you where on your computer a given executable lives.

```
> which python

> which python3
```

Running Python

Let's run python:

```
> python3

Python 3.8.2 (default, Feb 26 2020, 22:21:03)
[GCC 9.2.1 20200130] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Writing Python

Python is a programming language.

It's named after Monty Python and was created by Guido van Rossum in 1991.

There are two major versions that are still circulating: 2 & 3.

Python2 has been dying a long, slow death for many years now. We will be using Python3 for this course.

A python file

Create a new file in vscode and call it `hello.py`.

Write the following in the file:

```
message = 'Hello CodeOp!'
print(message)
```

A python file

In your terminal:

1. Navigate to the directory of the `hello.py` file you just created.
2. Run the file with python: `python3 hello.py`.

VSCode and Python

1. Install the "Python" extension (by Microsoft)
2. Install some python libraries on your computer (pip, pylint, pytest, pytest-xdist)