Computação Científica com Python

Paulo Henrique Junqueira Amorim

Divisão de Tecnologias Tridimensionais - Centro de Tecnologia da Informação
Renato Archer - CTI

Setembro de 2013

Sumário

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV
  - VTK - Visualization Toolkit

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV
  - VTK - Visualization Toolkit
  - ITK - Insight Segmentation and Registration Toolkit

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV
  - VTK - Visualization Toolkit
  - ITK - Insight Segmentation and Registration Toolkit
  - Mahotas(Visão Computacional)

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV
  - VTK - Visualization Toolkit
  - ITK - Insight Segmentation and Registration Toolkit
  - Mahotas(Visão Computacional)
  - GDCM - Grassroots DICOM

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV
  - VTK - Visualization Toolkit
  - ITK - Insight Segmentation and Registration Toolkit
  - Mahotas(Visão Computacional)
  - GDCM - Grassroots DICOM
  - PyDICOM

Introdução

- Existem diversas bibliotecas em Python para trabalhar em várias áreas da ciência.
  - Numpy - Numeric Python.
  - Scipy - Scientific python.
  - PIL - Python Imaging Library
  - Matplotlib
  - Scikit (learn, image ...)
  - Astropy
  - PyWavelets
  - Python - OpenCV
  - VTK - Visualization Toolkit
  - ITK - Insight Segmentation and Registration Toolkit
  - Mahotas(Visão Computacional)
  - GDCM - Grassroots DICOM
  - PyDICOM
  - ...

Introdução

Python - O criador...

Figura: Guido van Rossum - Holanda 1991

Introdução

Python - O nome...

Figura: Monty Python

Introdução

Python - Características

- Linguagem fácil e intuitiva.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.
- Desenvolvimento rápido.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.
- Desenvolvimento rápido.
- Interpretada.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.
- Desenvolvimento rápido.
- Interpretada.
- Imperativa.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.
- Desenvolvimento rápido.
- Interpretada.
- Imperativa.
- Orientada Objetos.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.
- Desenvolvimento rápido.
- Interpretada.
- Imperativa.
- Orientada Objetos.
- Tudo é um objeto.

Introdução

Python - Características

- Linguagem fácil e intuitiva.
- Código aberto.
- Sintaxe tão interlegivel quanto inglês.
- Desenvolvimento rápido.
- Interpretada.
- Imperativa.
- Orientada Objetos.
- Tudo é um objeto.
- Comunica com C/C++ através de Wrappers.

Introdução

Python - Tipos

- Números: Int, long, float, complex
- Booleano: True, False
- String: str, unicode
- Listas e Tuplas: list, tuple
- Dicionários: dict
- Arquivos: file
- Nulo: None

Introdução

Python - Operadores

- Comparação
  - $==, !=, <, <=, >, >=$
- Lógicos
  - and or not
- Matemáticos
  - $+, -, *, /, \%, **$
- Bit
  - $<<, >>, \hat{}, |, \tilde{}, \&$

Introdução

Python - Um pequeno exemplo

```
1  # -*- coding: UTF-8 -*-
2
3  print "Olá Mundo!!"
```

exemplos/ola.py

```
1  # -*- coding: UTF-8 -*-
2
3  linguagem = "python"
4
5  if linguagem == "python":
6      print "=)"
7  else:
8      print "(="
```

- Blocos são delimitados por 4 espaços ou 1 tab

exemplos/condicional.py

Introdução

Python - Laço de repetição

```
1  # -*- coding: UTF-8 -*-
2
3  for i in range(0,10):
4      print i
```

- de 0 (inclusive) até 10 ("exclusive")

exemplos/loop_for.py

```
1  # -*- coding: UTF-8 -*-
2  n = 0
3  while(n <= 10):
4          print n
5          n += 1
```

- Agora sim.. de 0 até 10.

exemplos/loop_while.py

Introdução

Python - Escrita de arquivo

```
1  # -*- coding: UTF-8 -*-
2
3  f = open("./teste.txt","w+")
4  for x in range(0,10):
5      f.write(str(x))
6  f.close()
```

exemplos/txt_escrita.py

Introdução

Python - Leitura de arquivo

```
1  # -*- coding: UTF-8 -*-
2
3  f = open("./teste.txt","r")
4  print f.read()
5  f.close()
```

exemplos/txt_leitura.py

```
1  # -*- coding: UTF-8 -*-
2
3  uf = ["BA","MG","SP", "RJ"]
4  l = [1, "MT", 0.4, [0,1,2]]
5
6  print uf[0]
```

- Listas são coleções de itens (objetos) que podem ter tipos misturados e mutaveis.
- Pode-se utilizar o índice para acessar.

exemplos/lista.py

```
1  # -*- coding: UTF-8 -*-
2
3  def soma(n1, n2):
4      return n1 + n2
5
6  print soma(5,10)
```

- Não é necessário especificar o tipo de retorno.

exemplos/func.py

Introdução

Python - Classe

```
1  # -*- coding: UTF-8 -*-
2  class Calculadora:
3      def __init__(self):
4          self.pi = 3.14
5
6      def soma(self, n1, n2):
7          return n1 + n2
8
9      def get_pi(self):
10         return self.pi
11
12 c = Calculadora()
13 print c.soma(2,2)
14 print c.get_pi()
15 print c.pi
```

exemplos/class.py

Numpy

Sobre

- É uma biblioteca com o básico de computação científica.
- `http://www.numpy.org/`
  - Matrizes N-Dimensão.
  - Algebra Linear.
  - Transformada de Fourier.
  - Estátistica Básica
  - ...

Numpy

Criando um vetor

```
1  # -*- coding: UTF-8 -*-
2
3  import numpy as np
4
5  v = np.array([5, 7, 7, 12])
```

exemplos/np_vetor.py

Numpy

Média

```
7  print "Média:",np.mean(v)
```

exemplos/np_vetor.py

Numpy

Mediana

```
8  print "Mediana:",np.median(v)
```

exemplos/np_vetor.py

Numpy

Percentil (Método do NIST)

- Dado: $v = [y_1, y_2, ..., y_n]$
- Calcular o rank do percentil: $rank = 1 + p(N - 1)$.
  - N é o tamanho do vetor.
  - p é o percentil.
- Dividir o valor do rank em parte inteira (i) e decimal (d).
- Se $i = 0$ então o valor é: $v[1]$
- Se $i = N$ então o valor é: $v[N]$
- Se $O < i < N$ então o valor é: $v[i] + d * (v[i + 1] - v[i])$

```
9  print "Percentil:",np.
       percentile(v, 25)
```

exemplos/np_vetor.py

Numpy

Criando uma matriz

$$M = \begin{bmatrix} 10 & 2 & 3 \\ -2 & 5 & 8 \\ 4 & 8 & -1 \end{bmatrix}$$

```
1  # -*- coding: UTF-8 -*-
2
3  import numpy as np
4
5  m = np.array([[10,2,3],[-2, 5,
       8],[4, 8,-1]])
```

exemplos/np_matriz.py

Numpy

Diagonal principal

$$\begin{bmatrix} 10 & 2 & 3 \\ -2 & 5 & 8 \\ 4 & 8 & -1 \end{bmatrix} = \begin{bmatrix} 10 & 5 & -1 \end{bmatrix}$$

```
9  print m.diagonal()
```

exemplos/np_matriz.py

Numpy

Matriz transposta

$$\begin{bmatrix} 10 & 2 & 3 \\ -2 & 5 & 8 \\ 4 & 8 & -1 \end{bmatrix}^T = \begin{bmatrix} 10 & -2 & 4 \\ 2 & 5 & 8 \\ 3 & 8 & -1 \end{bmatrix}$$

```
11  print m.transpose()
```

exemplos/np_matriz.py

Numpy

Determinante - pacote algebra linear

$$det \begin{bmatrix} 10 & 2 & 3 \\ -2 & 5 & 8 \\ 4 & 8 & -1 \end{bmatrix} = -738$$

```
13  print np.linalg.det(m)
```

exemplos/np_matriz.py

Numpy

Multiplicação de matrizes (numpy.array)

$$mr = \begin{bmatrix} 10 & 2 & 3 \\ -2 & 5 & 8 \\ 4 & 8 & -1 \end{bmatrix} * \begin{bmatrix} 12 & 2 & 3 \\ 4 & 35 & 6 \\ 8 & 10 & 7 \end{bmatrix}$$

```
1  # -*- coding: UTF-8 -*-
2
3  import numpy as np
4
5  m = np.array([[10,2,3],[-2, 5,
       8],[4, 8,-1]])
6  n = np.array
     ([[12,2,3],[4,35,6],[8,10,7]])
7
8  print np.dot(m,n)
```

exemplos/np_mult.py

Numpy

Multiplicação de matrizes (numpy.matrix)

$$mr = \begin{bmatrix} 10 & 2 & 3 \\ -2 & 5 & 8 \\ 4 & 8 & -1 \end{bmatrix} * \begin{bmatrix} 12 & 2 & 3 \\ 4 & 35 & 6 \\ 8 & 10 & 7 \end{bmatrix}$$

```
10  m = np.matrix([[10,2,3],[-2,
        5, 8],[4, 8,-1]])
11  n = np.matrix
        ([[12,2,3],[4,35,6],[8,10,7]])

12
13  print m * n
```

exemplos/np_mult.py

Scipy

Sobre

- Scipy contém rotinas númericas para:
  - Integral
  - Interpolação
  - Transformada de Fourier
  - Processamento de Sinais
  - Processamento de Imagens
  - Estatística
  - Entrada/Saída de arquivos
  - ...
- Trabalha com outras bibliotecas como numpy e matplotlib.
- `http://www.scipy.org/`

Scipy

Integral

$$\int_0^{10} f(x)dx$$

```
1  # -*- coding: UTF-8 -*-
2
3  import scipy.integrate as i
4
5  def f(x):
6      return x + 2
7
8  print i.quad(f,0,10)
```

exemplos/sp_integ.py

Scipy

Interpolação

```
1  # -*- coding: UTF-8 -*-
2
3  import scipy.interpolate as i
4  import numpy as np
5
6  x = np.linspace(0, 10, 6)
7  y = np.sin(x)
8
9  f = i.interp1d(x, y, kind='
       cubic')
10 xinterp = np.linspace(0, 10,
       100)
```

exemplos/sp_interp.py

Matplotlib

Sobre

- Contém ferramentas para criar e plotar gráficos e imagens.
- Recebe dados do numpy, scipy e outras bibliotecas.
- Suporta gráficos 2D e 3D.
- `http://matplotlib.org/`

Matplotlib

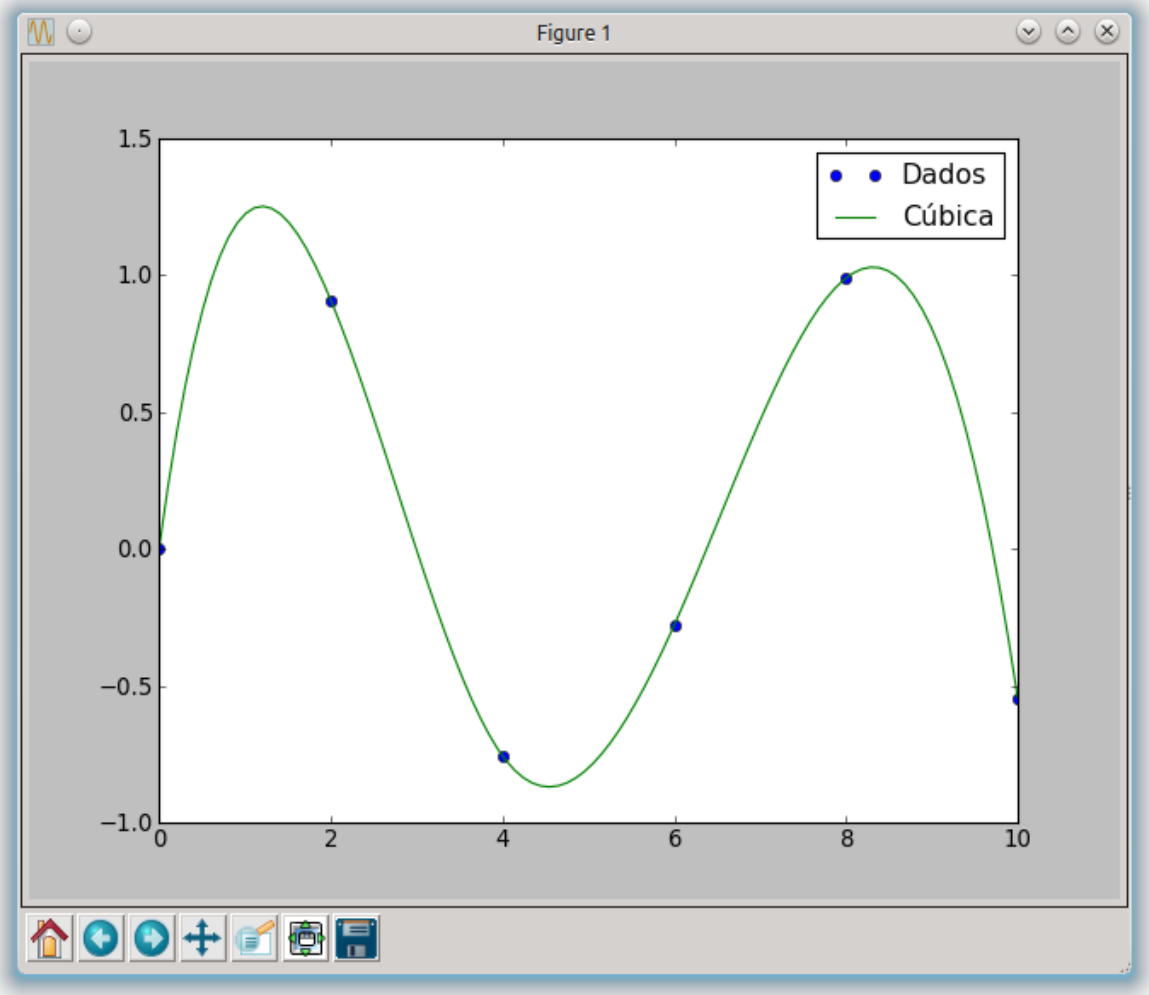
Gráfico da interpolação

```
12  import matplotlib.pyplot as
        plt
13  plt.plot(x,y,"o",xinterp,f(
        xinterp),"-")
14  plt.legend(["Dados", u"Cúbica"
        ],loc="best")
15  plt.show()
```

exemplos/sp_interp.py

Matplotlib

Gráfico da interpolação



Matplotlib

Gráfico de pizza

```
1  import matplotlib.pyplot as
       plt
2
3  labels =  '4 queijos', 'Frango
       /Catupiry', 'Portuguesa', '
       Provolone'
4  sizes = [10, 40, 45, 5]
5  colors = ['lightgreen','yellow
       ','lightskyblue', '
       lightgray']
6  explode = (0, 0, 0.1, 0)
```

exemplos/plot_pizza.py

Matplotlib

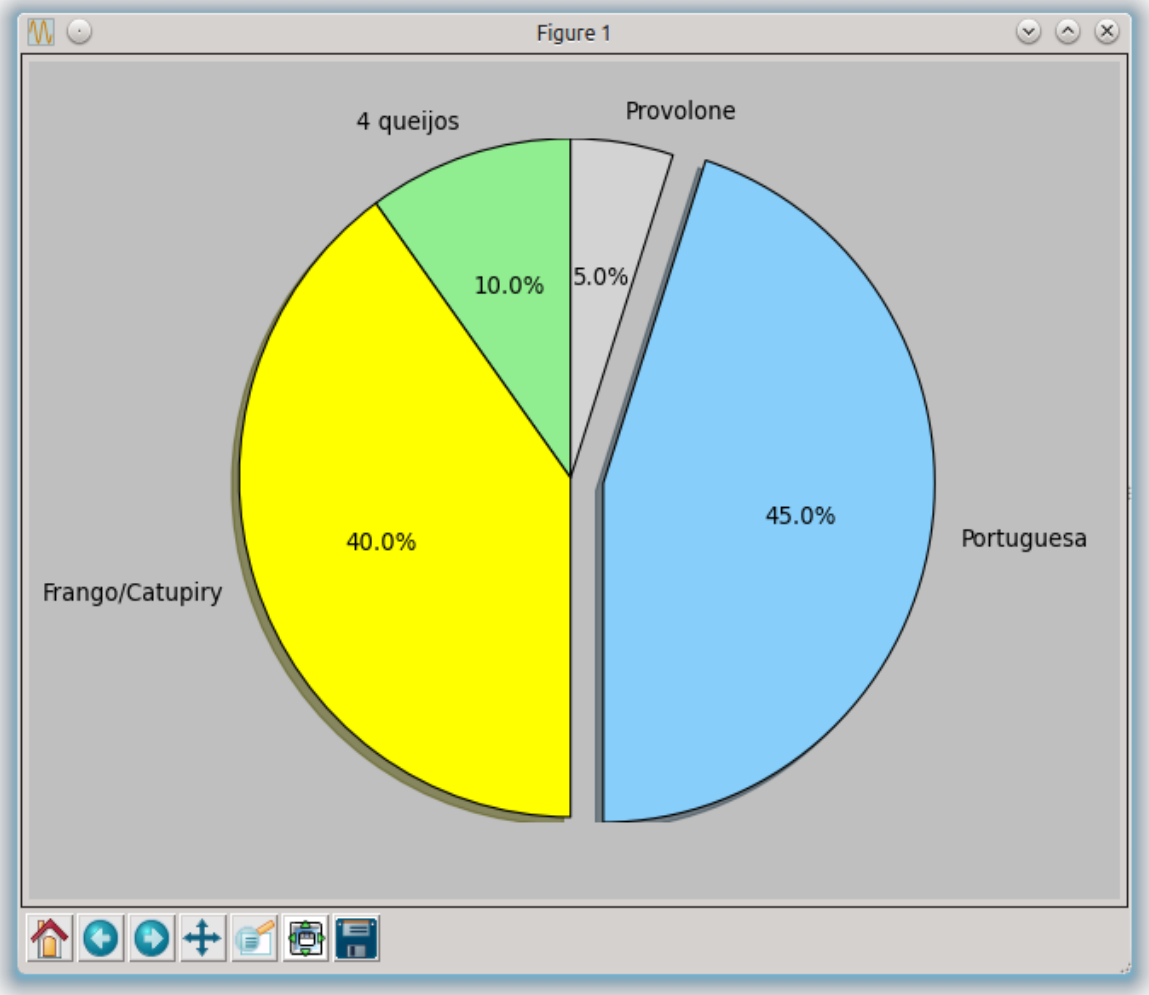
Gráfico de pizza

```
8  plt.pie(sizes, explode=explode
       , labels=labels, colors=
       colors, autopct='%1.1f%%',
       shadow=True, startangle=90)
9
10 plt.axis('equal')
11 plt.show()
```

exemplos/plot_pizza.py

Matplotlib

Gráfico de pizza



Objetivo:

- Criar uma matriz 100x100.
- Desenhar um círculo no centro.

Relembrando (Círculo):

$$x = xc + r * cos(\theta)$$
$$y = yc + r * sin(\theta)$$

Matplotlib

Plotagem de array

```
1  # -*- coding: UTF-8 -*-
2
3  import numpy as np
4  import matplotlib.pyplot as
       plt
5
6  img = np.zeros((100,100))
7
8  xc = 50
9  yc = 50
10 r = 40
```

exemplos/plot_array.py

Matplotlib

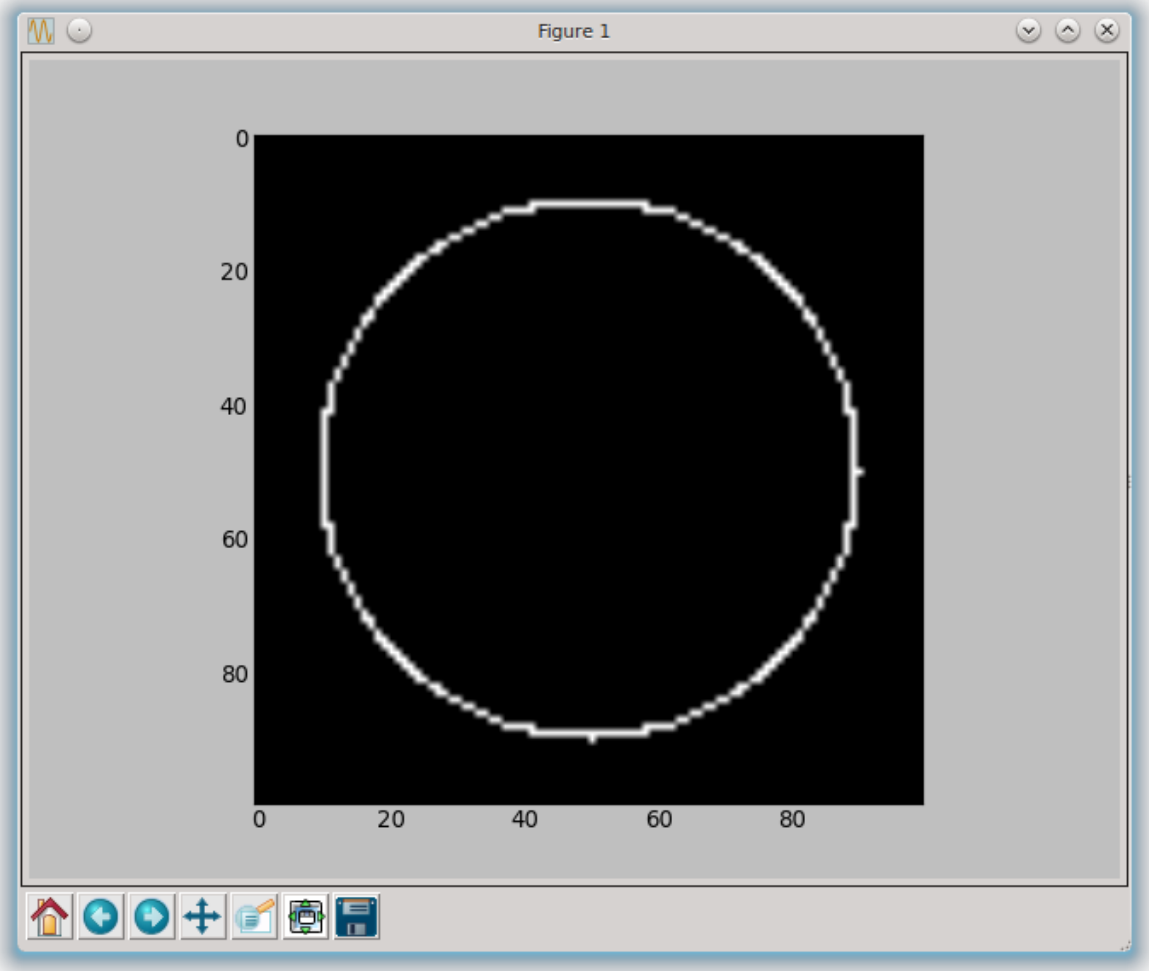
Plotagem de array

```
12  for circ in xrange(360):
13      theta = np.radians(circ)
14      x = xc + r*np.cos(theta)
15      y = yc + r*np.sin(theta)
16      img[x,y] = 255
17
18  plt.imshow(img, cmap=plt.cm.
        Greys_r)
19  plt.show()
```

exemplos/plot_array.py

Matplotlib

Plotagem de array



Matplotlib

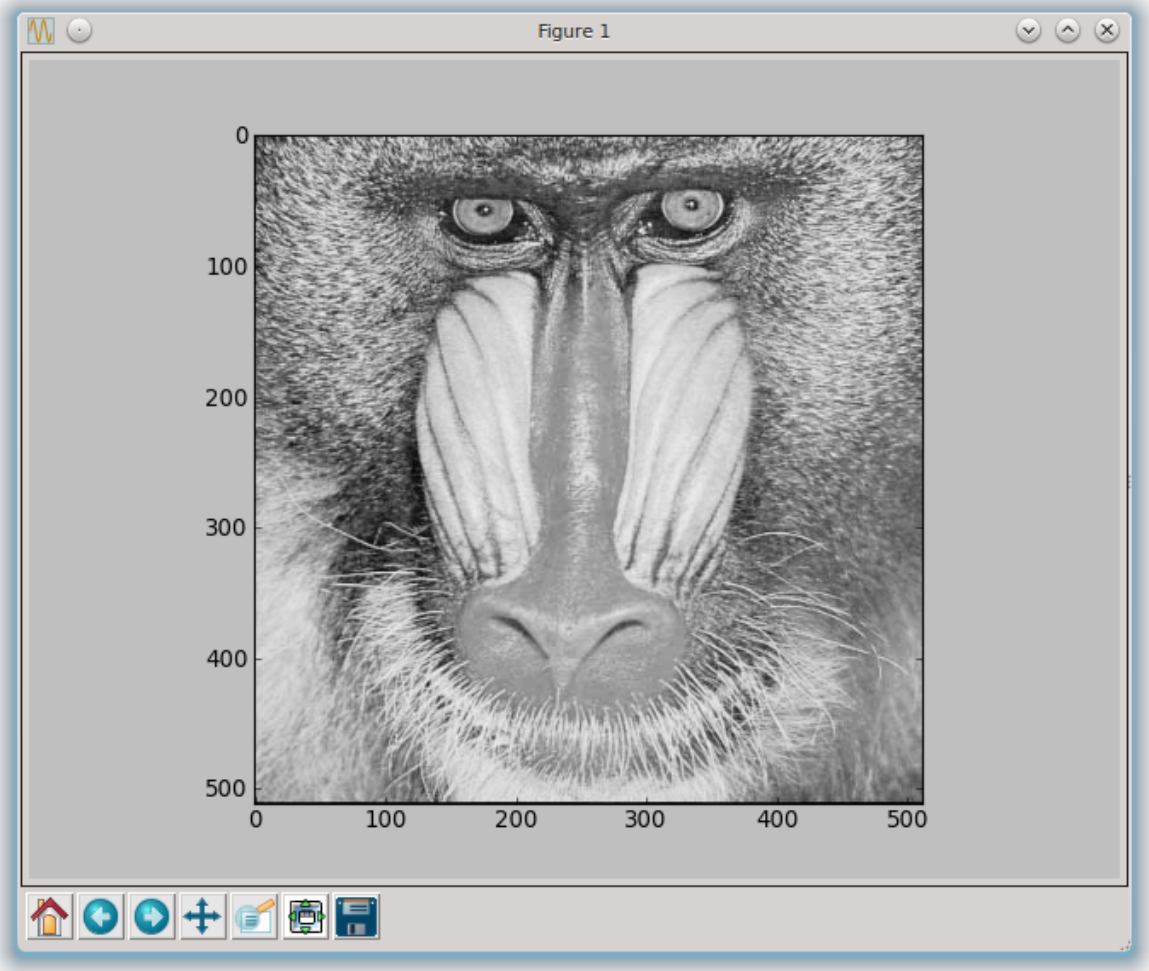
Plotagem de imagem

```
1  # -*- coding: UTF-8 -*-
2
3  import scipy.misc as misc
4  import matplotlib.pyplot as
       plt
5
6  img = misc.imread("./img/
       baboon.tif")
7  plt.imshow(img, cmap=plt.cm.
       Greys_r)
8  plt.show()
```

exemplos/plot_image.py

Matplotlib

Plotagem de imagem



Matplotlib

Plotagem de múltiplas imagens

```
1  # -*- coding: UTF-8 -*-
2
3  import scipy.misc as misc
4  import scipy.ndimage as ndi
5  import matplotlib.pyplot as
       plt
6
7  img = misc.imread("./img/
       baboon.tif")
8
9  img_filt = ndi.median_filter(
       img,(6,6))
```

exemplos/plot_med.py

Matplotlib

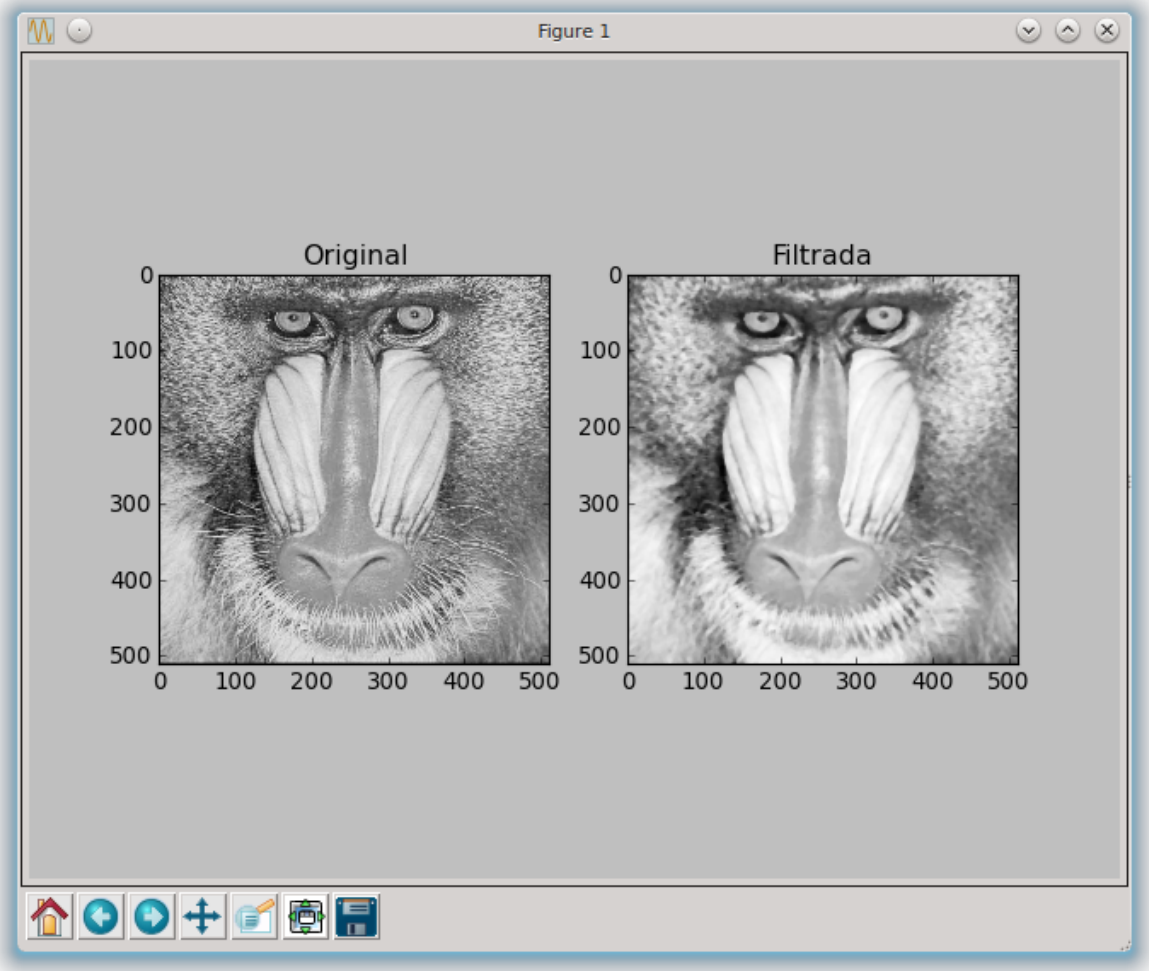
Plotagem de múltiplas imagens

```
11  fig = plt.figure()
12
13  fig.add_subplot(1,2,1)
14  plt.title("Original")
15  plt.imshow(img,cmap=plt.cm.
        Greys_r)
16
17  fig.add_subplot(1,2,2)
18  plt.title("Filtrada")
19  plt.imshow(img_filt, cmap=plt.
        cm.Greys_r)
20
21  plt.show()
```

exemplos/plot_med.py

Matplotlib

Plotagem de múltiplas imagens



Scikit-learn

Sobre

- Contém ferramentas para aprendizagem de máquina.
  - Supervisionada.
    - SVM (Máquina de vetores de suporte).
    - Gradiente descendente estocástico.
    - KNN (K-vizinhos mais próximos).
    - ...
  - Não supervisionada.
    - Clustering.
    - PCA (Análise de componentes principais)
    - Modelo oculto de Markov.
    - ...
- Suporta dados do numpy e scipy.
- `http://scikit-learn.org/`

Scikit-learn

Problema

- Dado dois conjuntos de dígitos (1 e 9) em duas classes respectivamente, predizer qual classe pertence um dado dígito. Os dígitos estão representados em forma de imagem.

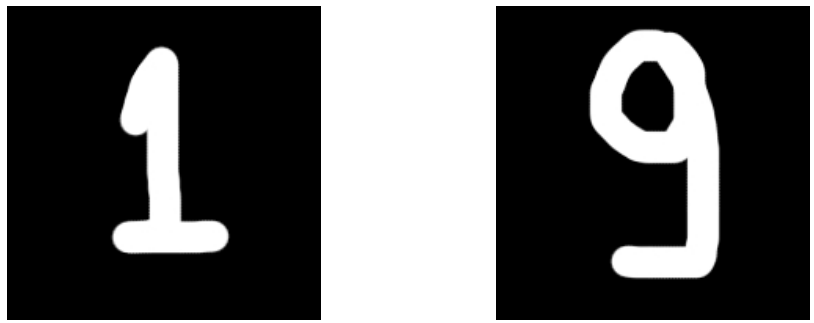


Figura: Um exemplo de cada classe.

Scikit-learn

Problema

- Dado dois conjuntos de dígitos (1 e 9) em duas classes respectivamente, predizer qual classe pertence um dado dígito. Os dígitos estão representados em forma de imagem.

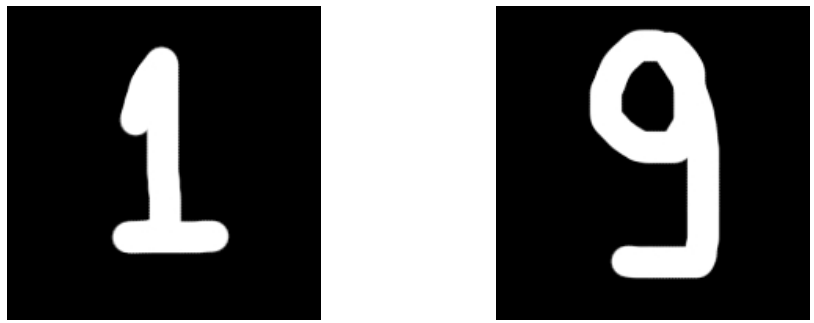


Figura: Um exemplo de cada classe.

- Podemos tentar resolver esse problema com descritores de imagem e o classificador SVM.

Scikit-learn

SVM

- Importar os módulos necessários.

```
1  # -*- coding: UTF-8 -*-
2
3  import glob
4  import scipy.misc as misc
5  import scipy.ndimage as ndi
6  import sklearn.svm as svm
7  import skimage.morphology as
       morp
```

exemplos/scikit_svm.py

Scikit-learn

SVM

- Inserir todas as imagens de treinamento em uma lista e ordena-las.

```
9  treinamento = glob.glob("img/
       treinamento/*.tif")
10 treinamento.sort()
```

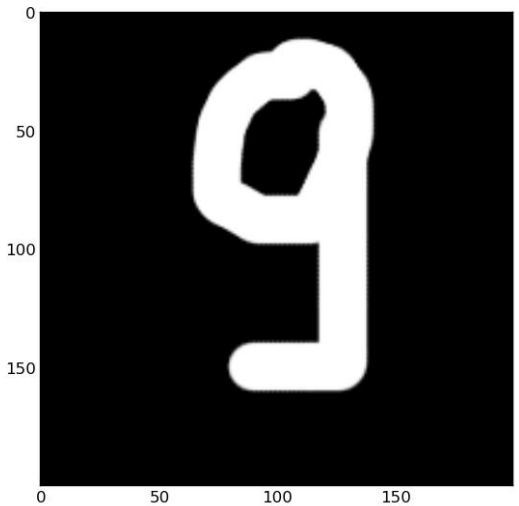
exemplos/scikit_svm.py

Scikit-learn

SVM

- Percorrer cada imagem, extrair características, normalizar e guardar em um vetor de treinamento.

```
12  caract = []
13  for f in treinamento:
14      img_entrada = misc.
            imread(f)
```



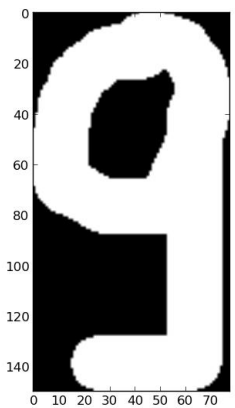
exemplos/scikit_svm.py

Scikit-learn

SVM

- Recortar somente o dígito de forma que fique em um retângulo envolvente.

```
15          rotulos, n_obj = ndi.
              label(img_entrada)
16          objetos = ndi.
              find_objects(rotulos)
17          img_objeto = rotulos[
              objetos[0]]
```



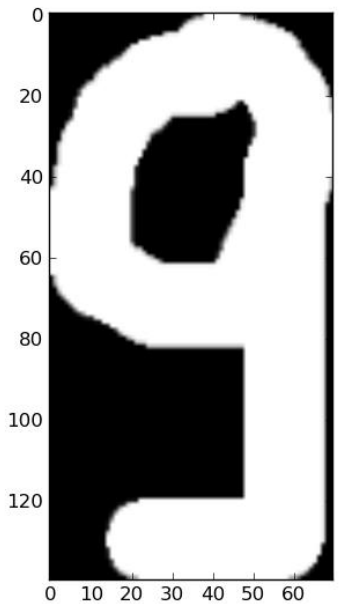
exemplos/scikit_svm.py

Scikit-learn

SVM

- Normalizar a imagem deixando todas com a mesma dimensão (140x70).

```
19          img_norm = misc.
               imresize(img_objeto
               ,(140,70))
```



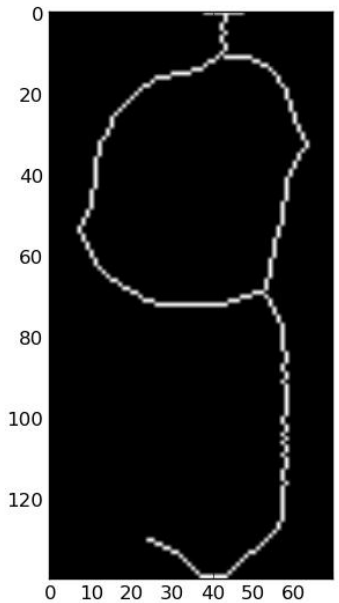
exemplos/scikit_svm.py

Scikit-learn

SVM

- Descrever o objeto por o seu eixo médio (conhecido também como descritor de esqueleto).

```
20          img_eixo = morp.
                medial_axis(img_norm)
```



exemplos/scikit_svm.py

Scikit-learn

SVM

- Transformar a matriz que representa a imagem em um vetor e adiciona-la no vetor de treinamento.

```
21          vetor_caract = img_eixo
                .reshape(140*70)
22          caract.append(
                vetor_caract)
```

exemplos/scikit_svm.py

Scikit-learn

SVM

- Indicar cada item do vetor de treinamento com o seu respectivo rótulo.

```
24  objetivo = ["1"] * 10 + ["9
      "] * 10
25
26  clf = svm.SVC()
27  clf.fit(caract,objetivo)
```

exemplos/scikit_svm.py

Scikit-learn

SVM

- Executar o mesmo passo anterior, mas para uma imagem de teste.

```
29  img_teste = misc.imread("
      img/teste/9a.tif")
30
31  rotulos, n_obj = ndi.label(
      img_teste)
32  objetos = ndi.find_objects(
      rotulos)
33  img_objeto = rotulos[
      objetos[0]]
```

exemplos/scikit_svm.py

Scikit-learn

SVM

```
35  img_norm = misc.imresize(
        img_objeto,(140,70))
36  img_eixo = morp.medial_axis(
        img_norm)
37
38  vetor_caract = img_eixo.
        reshape(140*70)
```

exemplos/scikit_svm.py

Scikit-learn

SVM

- Resultado.

```
39  pred = clf.predict(
        vetor_caract)
40
41  print "\n\n O numero de
        entrada é", pred[0]
```

exemplos/scikit_svm.py

FIM

Contato

- paulo.amorim @ cti.gov.br
- paulojamorim @ gmail.com
- `http://github.com/paulojamorim`