

【機器學習】【SVD-3】SVD降維的應用簡介+ 降維示例展示+ Python代碼實現

原創 CV_ML_DP 最後發佈於2018-05-24 18:05:46 閱讀數3935 收藏 ☆

🔖 2

🔗 展開

💬 6

📖

☆

📱

<

>

👍

1.SVD降維的基本原理

SVD降維的基本原理，可以詳見以前文章：<https://blog.csdn.net/u012421852/article/details/80433463>

2.降維示例展示

降維的理論以及和意義不再贅述，此處僅僅給出SVD分解降維的一個應用示例。

Step1:準備要降維的數據矩陣M，以及奇異值開方和占比閾值percentage

```
1 data = np.array([[5, 5, 0, 5],
2                  [5, 0, 3, 4],
3                  [3, 4, 0, 3],
4                  [0, 0, 5, 3],
5                  [5, 4, 4, 5],
6                  [5, 4, 5, 5]])
7 percentage = 0.9
```

Step2:對數據矩陣M進行奇異值分解

```
(U, S, VT) = np.linalg.svd(M)
```

解釋np.linalg.svd(M)的返回值U，S，VT依次對應於下面SVD公式中的U，Σ，VT，

注意：svd()方法返回的第三個值是SVD中的VT，而不是V，後面降維時直接使用VT[:K, :len(VT)]作為降維後的SVD公式中的VT

注意：S是M的所有奇異值的數組

$$Data_{mxn} = U_{mxm} \Sigma_{mxn} V_{nxn}^T$$

求得的U，S，VT如下所示：

```
1 降维前的U,S,VT依次为:
2 (6, 6) U:
3 [[-0.44721867  0.53728743  0.00643789 -0.50369332 -0.38572204 -0.32982993]
4  [-0.35861531 -0.24605053 -0.86223083 -0.14584826  0.07797125  0.20015231]
5  [-0.29246336  0.40329582  0.22754042 -0.10376096  0.4360044  0.70652449]
6  [-0.20779151 -0.67004393  0.3950621  -0.58878098  0.02599042  0.06671744]
7  [-0.50993331 -0.05969518  0.10968053  0.28687443  0.59460659 -0.53714128]
8  [-0.53164501 -0.18870999  0.19141061  0.53413013 -0.54845844  0.24290419]]
9 (4,) S:
10 [ 17.71392084  6.39167145  3.09796097  1.32897797]
11 (4, 4) VT:
12 [[-0.57098887 -0.4274751  -0.38459931 -0.58593526]
13  [ 0.22279713  0.51723555 -0.82462029 -0.05319973]
14  [-0.67492385  0.69294472  0.2531966  -0.01403201]
15  [ 0.41086611  0.26374238  0.32859738 -0.80848795]]
```

Step3:根據percentage求k值

```
1 def _calc_k(self, percentge):
2     '''确定k值: 前k个奇异值的平方和占比 >=percentage，求满足此条件的最小k值
3     :param percentage, 奇异值平方和的占比的閾值
4     :return 满足閾值percentage的最小k值
5     '''
6     self.k = 0
7     #用户数据矩阵的奇异值序列的平方和
8     total = sum(np.square(self.S))
9     svss = 0 #奇异值平方和 singular values square sum
10    for i in range(np.shape(self.S)[0]):
11        svss += np.square(self.S[i])
12        if (svss/total) >= percentge:
13            self.k = i+1
14            break
15    return self.k
```

🔊

舉報

求得的k值為： 2

即成立： 前2個奇異值平方和/ 所有奇異值平方和> = 0.9

Step4：根據求得的k構造k階奇異值對角陣

```
1         def _buildSD(self, k):
2             '''构建由奇异值组成的对角矩阵
3             :param k, 根据奇异值开放和的占比阈值计算出来的k值
4             :return 由k个前奇异值组成的对角矩阵
5             '''
6             #方法1：用数组乘方法
7             self.SD = np.eye(self.k) * self.S[:self.k]
8
9             #方法2：用自定义方法
10            e = np.eye(self.k)
11            for i in range(self.k):
12                e[i,i] = self.S[i]
13
14            return self.SD
```

Step5：根據求得的k求降維後的U和VT

```
1 new_U  = U[:len(U), :k]
2 new_VT = VT[:k, :len(VT)]
```

k=2時，降維後的U和VT為：

```
1 降维后的U，VT依次为：
2 (6,) new_U=U[:,2]:
3 [[-0.44721867  0.53728743]
4  [-0.35861531 -0.24605053]
5  [-0.29246336  0.40329582]
6  [-0.20779151 -0.67004393]
7  [-0.50993331 -0.05969518]
8  [-0.53164501 -0.18870999]]
9
10 (2, 4) new_VT=VT[:,4]:
11 [[-0.57098887 -0.4274751  -0.38459931 -0.58593526]
12  [ 0.22279713  0.51723555 -0.82462029 -0.05319973]]
```

Step6：用降維後的U，SD，VT求降維後的用戶數據矩陣new_M

記原始用戶數據矩陣的被評價的商品數為n，用戶數為m降維後的SVD公式稱為：

new_M = U(n,k)SD(k,k)VT(k,m)

注：U,SD,VT之間是矩陣乘操作np.dot()，不是矩陣點乘np.multiply()

求得降維後的用戶數據矩陣為：


```
1 用户对商品的评分矩阵M（4个用户对6个商品的评分矩阵）：
2 [[5 5 0 5] #所有用户对商品1的评分值序列
3  [5 0 3 4] #所有用户对商品2的评分值序列
4  [3 4 0 3] #所有用户对商品3的评分值序列
5  [0 0 5 3] #所有用户对商品4的评分值序列
6  [5 4 4 5] #所有用户对商品5的评分值序列
7  [5 4 5 5]] #所有用户对商品6的评分值序列
8 降维后的评分矩阵new_M:
9  [[ 5.28849359  5.16272812  0.21491237  4.45908018]
10  [ 3.27680994  1.90208543  3.74001972  3.80580978]
11  [ 3.53241827  3.54790444 -0.13316888  2.89840405]
12  [ 1.14752376 -0.64171368  4.94723586  2.3845504 ]
13  [ 5.07268706  3.66399535  3.78868965  5.31300375]
14  [ 5.10856595  3.40187905  4.6166049  5.58222363]]
```


可以直觀會發現，降維後的數據矩陣new_M和原始用戶數據矩陣M很接近， new_M能夠表達原始數據矩陣M的中所有數據的相關性


也會發現， SVD降維是一種數據的有損壓縮。


3.Python代碼實現（Release版本）


3.1代碼


2





6


















舉報


此代碼是降維應用實例的精簡代碼，如果是理解SVD分解降維示例的數學求解過程，可以看4節給出的Debug版本代碼。


人肉出品，代碼詳見如下：


```
1  # -*- coding: utf-8 -*-
2  """
3  @author: 蔚藍的天空Tom
4  Talk is cheap, show me the code
5  Aim:svd分解降维应用示例的代码实现(Release版本)
6  CSDN URL:https://mp.csdn.net/postedit/80450590
7  """
8
9  import numpy as np
10
11 class CSVD(object):
12     '''
13     实现svd分解降维应用示例的Python代码
14     '''
15
16     def __init__(self, data):
17         self.data = data          #用户数据
18         self.S = []              #用户数据矩阵的奇异值序列 singular values
19         self.U = []              #svd后的单位正交向量
20         self.VT = []             #svd后的单位正交向量
21         self.k = 0               #满足self.p的最小k值(k表示奇异值的个数)
22         self.SD = []            #对角矩阵, 对角线上元素是奇异值 singular values diagonal matrix
23
24     def _svd(self):
25         '''
26         用户数据矩阵的svd奇异值分解
27         '''
28         self.U, self.S, self.VT = np.linalg.svd(self.data)
29         return self.U, self.S, self.VT
30
31     def _calc_k(self, percentge):
32         '''确定k值: 前k个奇异值的平方和占比 >=percentage, 求满足此条件的最小k值
33         :param percentage, 奇异值平方和的占比的閾值
34         :return 满足閾值percentage的最小k值
35         '''
36         self.k = 0
37         #用户数据矩阵的奇异值序列的平方和
38         total = sum(np.square(self.S))
39         svss = 0 #奇异值平方和 singular values square sum
40         for i in range(np.shape(self.S)[0]):
41             svss += np.square(self.S[i])
42             if (svss/total) >= percentge:
43                 self.k = i+1
44                 break
45         return self.k
46
47     def _buildSD(self, k):
48         '''构建由奇异值组成的对角矩阵
49         :param k,根据奇异值开放和的占比閾值计算出来的k值
50         :return 由k个前奇异值组成的对角矩阵
51         '''
52         #方法1: 用数组乘方法
53         self.SD = np.eye(self.k) * self.S[:self.k]
54
55         #方法2: 用自定义方法
56         e = np.eye(self.k)
57         for i in range(self.k):
58             e[i,i] = self.S[i]
59
60         return self.SD
61
62     def DimReduce(self, percentage):
63         '''
64         SVD降维
65         :param percentage, 奇异值开方和的占比閾值
66         :return 降维后的用户数据矩阵
67         '''
68         #Step1:svd奇异值分解
69         self._svd()
70         #Step2:计算k值
71         self._calc_k(percentge)
72         print('\n按照奇异值开方和占比閾值percentage=%d, 求得降维的k=%d'%(percentage, self.k))
73         #Step3:构建由奇异值组成的对角矩阵singular values diagonal
74         self._buildSD(self.k)
75         k,U,SD,VT = self.k,self.U, self.SD, self.VT
76         #Step4:按照svd分解公式对用户数据矩阵进行降维, 得到降维压缩后的数据矩阵
```


2





6




















舉報


```
77 |
78 |         a = U[:len(U), :k]
79 |     b = np.dot(SD, VT[:k, :len(VT)])
80 |     newData = np.dot(a,b)
81 |     return newData
82 |
83 | def CSVD_manual():
84 |     ##训练数据集，用户对商品的评分矩阵，行为多个用户对单个商品的评分，列为用户对每个商品的评分
85 |     data = np.array([[5, 5, 0, 5],
86 |                      [5, 0, 3, 4],
87 |                      [3, 4, 0, 3],
88 |                      [0, 0, 5, 3],
89 |                      [5, 4, 4, 5],
90 |                      [5, 4, 5, 5]])
91 |
92 |     percentage = 0.9
93 |     svdor = CSVD(data)
94 |     ret = svdor.DimReduce(percentage)
95 |     print('=====')
96 |     print('原始用户数据矩阵:\n', data)
97 |     print('降维后的数据矩阵:\n', ret)
98 |     print('=====')
99 |
100 | if __name__=='__main__':
    CSVD_manual()
```


2





6














3.2運行結果

```
1 | runfile('C:/Users/tom/svd_reduceDim_release.py', wdir='C:/Users/tom')
2 |
3 | 按照奇异值开方和占比阈值percentage=0，求得降维的k=2
4 | =====
5 | 原始用户数据矩阵:
6 | [[5 5 0 5]
7 |  [5 0 3 4]
8 |  [3 4 0 3]
9 |  [0 0 5 3]
10 | [5 4 4 5]
11 | [5 4 5 5]]
12 | 降维后的数据矩阵:
13 | [[ 5.28849359  5.16272812  0.21491237  4.45908018]
14 |  [ 3.27680994  1.90208543  3.74001972  3.80580978]
15 |  [ 3.53241827  3.54790444 -0.13316888  2.89840405]
16 |  [ 1.14752376 -0.64171368  4.94723586  2.3845504 ]
17 |  [ 5.07268706  3.66399535  3.78868965  5.31300375]
18 |  [ 5.10856595  3.40187905  4.6166049  5.58222363]]
19 | =====
```

4.Python代碼實現（debug版本）

4.1代碼


注：此代碼為展示示例的數學求解過程的代碼，不是精簡代碼


```
1 | # -*- coding: utf-8 -*-
2 | """
3 | @author: 蔚藍的天空Tom
4 | Talk is cheap, show me the code
5 | Aim:svd分解降维应用示例的代码实现
6 | CSDN URL:https://mp.csdn.net/postedit/80450590
7 | """
8 |
9 | import numpy as np
10 | from numpy import linalg as LA
11 |
12 | class CSVD(object):
13 |     '''
14 |     实现SVD分解降维应用示例的数学求解过程的Python代码
15 |     '''
16 |
17 |     def __init__(self, data):
18 |         self.data = data          #用户数据
19 |         self.S = []              #用户数据矩阵的奇异值序列 singular values
20 |         self.U = []              #svd后的单位正交向量
21 |         self.VT = []             #svd后的单位正交向量
22 |         self.k = 0                #满足self.p的最小k值(k表示奇异值的个数)
23 |         self.SD = []             #对角矩阵，对角线上元素是奇异值 singular values diagonal matrix
24 |
```





舉報


```
25 #svd奇异值分解
26         self._svd()
27
28 def _svd(self):
29     '''
30     用户数据矩阵的svd奇异值分解
31     '''
32     u,s,v = np.linalg.svd(self.data)
33     (self.U, self.S, self.VT) = (u, s, v)
34     return self.U, self.S, self.VT
35
36 def _calc_k(self, percentge):
37     '''确定k值: 前k个奇异值的平方和占比 >=percentage, 求满足此条件的最小k值
38     :param percentage, 奇异值平方和的占比的阈值
39     :return 满足阈值percentage的最小k值
40     '''
41     self.k = 0
42     #用户数据矩阵的奇异值序列的平方和
43     total = sum(np.square(self.S))
44     svss = 0 #奇异值平方和 singular values square sum
45     for i in range(np.shape(self.S)[0]):
46         svss += np.square(self.S[i])
47         if (svss/total) >= percentge:
48             self.k = i+1
49             break
50     return self.k
51
52 def _buildSD(self, k):
53     '''构建由奇异值组成的对角矩阵
54     :param k,根据奇异值开放和的占比阈值计算出来的k值
55     :return 由k个前奇异值组成的对角矩阵
56     '''
57     #方法1: 用数组乘方法
58     self.SD = np.eye(self.k) * self.S[:self.k]
59
60     #方法2: 用自定义方法
61     e = np.eye(self.k)
62     for i in range(self.k):
63         e[i,i] = self.S[i]
64
65     return self.SD
66
67 def DimReduce(self, percentage):
68     '''
69     SVD降维
70     :param percentage, 奇异值开方和的占比阈值
71     :return 降维后的用户数据矩阵
72     '''
73     #计算k值
74     self._calc_k(percentge)
75     print('\n按照奇异值开方和占比阈值percentage=%d, 求得降维的k=%d'%(percentage, self.k))
76     #构建由奇异值组成的对角矩阵singular values diagonal
77     self._buildSD(self.k)
78     k,U,SD,VT = self.k,self.U, self.SD, self.VT
79     #按照svd分解公式对用户数据矩阵进行降维, 得到降维压缩后的数据矩阵
80     print('\n降维前的U,S,VT依次为:')
81     print(np.shape(U), 'U:\n', U)
82     print(np.shape(self.S), 'S:\n', self.S)
83     print(np.shape(VT), 'VT:\n', VT)
84     print('\n降维后的U,SD,VT依次为:')
85     print(np.shape(U[:len(U),k]), 'U=U[:%d,:%d]:\n'%(len(U),k), U[:len(U), :k])
86     print(np.shape(SD), 'SD=SD[:%d, :%d]:\n'%(k,k), SD[:k, :k])
87     print(np.shape(VT[:k, :len(VT)]), 'VT=VT[:%d, :%d]:\n'%(k, len(VT)), VT[:k, :len(VT)])
88
89     a = U[:len(U), :k]
90     b = np.dot(SD, VT[:k, :len(VT)])
91     newData = np.dot(a,b)
92     return newData
93
94 def CSVD_manual():
95     ##训练数据集, 用户对商品的评分矩阵, 行为多个用户对单个商品的评分, 列为用户对每个商品的评分
96     data = np.array([[5, 5, 0, 5],
97                      [5, 0, 3, 4],
98                      [3, 4, 0, 3],
99                      [0, 0, 5, 3],
100                     [5, 4, 4, 5],
101                     [5, 4, 5, 5]])
102     percentage = 0.9
103     svdor = CSVD(data)
104     ret = svdor.DimReduce(percentge)
```


2





6
















舉報

```
105 |         print('=====')106 |         print('原始用户数据矩阵:\n', data)
107 |         print('降维后的数据矩阵:\n', ret)
108 |         print('=====')
109 |
110 | if __name__ == '__main__':
111 |     CSVD_manual()
```

4.2運行結果

```
1 | 按照奇异值开方和占比阈值percentage=0，求得降维的k=2
2 |
3 | 降维前的U,S,VT依次为：
4 | (6, 6) U:
5 | [[-0.44721867  0.53728743  0.00643789 -0.50369332 -0.38572204 -0.32982993]
6 | [-0.35861531 -0.24605053 -0.86223083 -0.14584826  0.07797125  0.20015231]
7 | [-0.29246336  0.40329582  0.22754042 -0.10376096  0.4360044  0.70652449]
8 | [-0.20779151 -0.67004393  0.3950621  -0.58878098  0.02599042  0.06671744]
9 | [-0.50993331 -0.05969518  0.10968053  0.28687443  0.59460659 -0.53714128]
10 | [-0.53164501 -0.18870999  0.19141061  0.53413013 -0.54845844  0.24290419]]
11 | (4,) S:
12 | [ 17.71392084   6.39167145   3.09796097   1.32897797]
13 | (4, 4) VT:
14 | [[-0.57098887 -0.4274751  -0.38459931 -0.58593526]
15 | [ 0.22279713  0.51723555 -0.82462029 -0.05319973]
16 | [-0.67492385  0.69294472  0.2531966  -0.01403201]
17 | [ 0.41086611  0.26374238  0.32859738 -0.80848795]]
18 |
19 | 降维后的U,SD,VT依次为：
20 | (6,) U=U[:, :2]:
21 | [[-0.44721867  0.53728743]
22 | [-0.35861531 -0.24605053]
23 | [-0.29246336  0.40329582]
24 | [-0.20779151 -0.67004393]
25 | [-0.50993331 -0.05969518]
26 | [-0.53164501 -0.18870999]]
27 | (2, 2) SD=SD[:, :2]:
28 | [[ 17.71392084   0.
29 |    0.         6.39167145]]
30 | (2, 4) VT=VT[:, :4]:
31 | [[-0.57098887 -0.4274751  -0.38459931 -0.58593526]
32 | [ 0.22279713  0.51723555 -0.82462029 -0.05319973]]
33 | =====
34 | ===原始用户数据矩阵：
35 | [[5 5 0 5]
36 | [5 0 3 4]
37 | [3 4 0 3]
38 | [0 0 5 3]
39 | [5 4 4 5]
40 | [5 4 5 5]]
41 | ===降维后的数据矩阵：
42 | [[ 5.28849359  5.16272812  0.21491237  4.45908018]
43 | [ 3.27680994  1.90208543  3.74001972  3.80580978]
44 | [ 3.53241827  3.54790444 -0.13316888  2.89840405]
45 | [ 1.14752376 -0.64171368  4.94723586  2.3845504 ]
46 | [ 5.07268706  3.66399535  3.78868965  5.31300375]
47 | [ 5.10856595  3.40187905  4.6166049  5.58222363]]
48 | =====
```

參考文獻：

[1][機器學習筆記]奇異值分解SVD簡介及其在推薦系統中的簡單應用

[2]【機器學習】推薦系統、SVD分解降維

[3] SVD在推薦系統中的應用

[4] SVD Recommendation System in Ruby

[5] 矩陣特徵值分解與奇異值分解含義解析及應用

[6] We Recommend a Singular Value Decomposition

(end)

👍 2

🔗

💬 6

📖

☆

📱

<

>

👑

🔄

舉報



CV_ML_DP

發布了248篇原創文章 · 獲贊206 · 訪問量43萬+

他的留言板

關注



想對作者說點什麼



2



6



查看



閱讀



SVD推薦算法python 實現

經典的SVD算法，預測評分的的計算公式如下：其中μ為評分的平均值，分別表示u用戶的偏置量和i物品的偏置量。...

博文

akiyamamio11的博客

來自：

機器學習_用SVD奇異值分解給數據降維

閱讀數 7144

若將矩陣看做一個變換，那麼任何這樣的變換可以看做是兩個旋轉和一個縮放變換的複合，這點和特徵值分解基本一...

博文

謝彥的技術博客

來自：

SVD 降維（相當好理解的文章）

閱讀數 1925

作者：belupreda鏈接：https://www.zhihu.com/question/34143886/answer/196294308來源：知乎著作權歸作...

博文

三眼二郎

來自：

PYTHON 機器學習實戰—— SVD奇異值分解數據降維

閱讀數 3260

PYTHON機器學習實戰——SVD奇異值分解數據降維博文末尾支持二維碼讚賞哦#-*- coding:utf-8 -*-#!/usr/bin/py...

博文

萬有文的博客

來自：

數據降維方法及Python 實現

閱讀數 4750

一、數據降維了解1.1、數據降維原理：機器學習領域中所謂的降維就是指採用某種映射方法，將原高維空間中的數據...

博文

yawei的博客

來自：

利用矩陣奇異值分解(SVD)進行降維

閱讀數 1043

一、SVD的优缺点及应用场合1.优点：简化数据，去除噪声，提高算法的结果2.缺点：数据的转换可能难以理解3.适用...

博文

来自：Curry_Coder的个...

推荐系统Koren’ s SVD++ Python实现

12-19

Netflix Prize上的奇异矩阵分解算法，在数据稀疏的时候可以较好的实现算法稳定性。http://gustavonarea.net/blog/po...

下載

svd降维与kmeans聚类

12-14

给公司做报告时写的ppt，希望大家能喜欢

下載

《机器学习实战笔记--降维技术 利用SVD来简化数据》

阅读数 336

奇异值分解：SVD（singular value decomposition）1.1 SVD的应用 利用SVD我们能用小的大数据来表示原始...

博文

来自：qq_41635352的博客

Python - 降维（PCA、核PCA、SVD、高斯随机映射 和 NMF）

阅读数 3272

以下内容来自《Python数据科学指南》降维方法比较：PCA：计算代价高昂，特征向量得存在线性相关。核PCA：特...

博文

来自：Alvin AI Blog

推荐系统相关embedding：SVD、SVD++

阅读数 321

继续讲embedding相关的一些东西，之前在公众号的地址：推荐系统相关embedding：SVD、SVD++欢迎关注我...

博文

来自：ty44111144ty的博客



akiyamamio11

16篇文章

关注

排名:千里之外



xieyan0811

331篇文章

关注

排名:1000+



三眼二郎

135篇文章

关注

排名:千里之外



EwenWanW

82篇文章

关注

排名:千里之外

nginx学习，看这一篇就够了：下载、安装。使用：正向代理、反向代理、负载均衡。常用命令和配置文件

阅读数 5万+

文章目录前言一、nginx简介1. 什么是 nginx 和可以做什么事情2.Nginx 作为 web 服务器3. 正向代理4. 反向代理5. ...

博文

来自：冯安晨

利用PCA，SVD原理实现对图片的特征提取（矩阵降维）

阅读数 308

关于PCA和SVD的区别，以及简单的应用实例简述SVD和PCA PCA SVD 区别功能快捷键合理的创建...

博文

来自：qq_43337858

PCA与SVD降维的结果比较（以Iris数据集为例）（附Python代码）

阅读数 308

比较结果左上：原始数据的前二维特征。右上：利用Sklearn库函数decomposition.PCA()对原数据进行PCA后，取...

博文

来自：koko_TT的博客

常见的降维方法（PCA，SVD）

阅读数 330

1、PCA降维（主成分分析）PCA降维就是去除线性相关，使得最后剩余的属性维度全都线性无关。其实：PCA降维...

博文 来自： [liymgf](#)

机器学习Python实现 SVD 分解

这篇文章主要是结合机器学习实战将推荐算法和SVD进行相应的结合任何一个矩阵都可以分解为SVD的形式其实SVD...

博文 来自： [xietingcanc](#)

网页实现一个简单的音乐播放器（大佬别看。（๑_๑））

今天闲着无事，就想写点东西。然后听了下歌，就打算写个播放器。于是乎用h5 audio的加上js简单的播放器完工了...

博文 来自： [qq_44210563](#)

【机器学习】【SVD-5】SVD在推荐（策略：TopK）系统中的应用简介 + 示例展示 + Python代码实现

SVD在推荐（策略：TopK）系统中的应用简介 + 示例展示 + Python代码实现

博文 来自： [蔚蓝的天空To](#)

史上最详细的IDEA优雅整合Maven+SSM框架（详细思路+附带源码）

网上很多整合SSM博客文章并不能让初探ssm的同学思路完全的清晰，可以试着关掉整合教程，摇两下头骨，哈一大...

博文 来自： [程序员宜春的](#)

【机器学习】【SVD-4】SVD在推荐（策略：TopOne）系统中的应用简介 + 示例展示 + Python代码实现

1.SVD降维的应用简介2.降维示例展示3.Python代码实现参考文献：[1][机器学习笔记]奇异值分解SVD简介及其在推...

博文 来自： [蔚蓝的天空To](#)

Reducing the Dimensionality of Data with Neural Networks：神经网络用于降维

阅读数 1万+

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural netw...

博文 来自： [wishchinYang的专...](#)

奇异值分解（SVD）与降维

阅读数 2379

一、特征分解 许多数学对象可以通过将它们分解成多个组成部分或者找到它们地 一些属性来更好的理解。这些属性...

博文 来自： [lpty的博客](#)

【机器学习】推荐系统、SVD分解降维

阅读数 9

推荐系统： 1.基于内容的实现：KNN等2.基于协同滤波(CF)实现： SVD→ pLSA(从LSA发展而来,由SVD实现)、LDA、 ...

博文 来自： [weixin_30730053...](#)

机器学习SVD的实例Python版本

阅读数 92

```
#!/usr/bin/python# -*- coding:utf-8 -*-import numpy as npimport osfrom PIL import Imageimport matpl...
```

博文 来自： [庄小焱](#)

SVD理论以及Python实现

阅读数 208

SVD将一个矩阵分解为U，V（U,V均为列正交矩阵，即列向量直接内积为0），中间的矩阵为对角阵，元素为奇异值...

博文 来自： [肥宅Sean](#)

t-SNE降维原理及python实现代码

阅读数 4367

t-SNE（Stochastic Neighbor Embedding t分布随机邻域嵌入） PCA的局限性PCA是一种线性算法，它不能解释...

博文 来自： [qq_27802435的博客](#)

奇异值分解(SVD)原理与在降维中的应用

阅读数 70

转自 <https://www.cnblogs.com/pinard/p/6251584.html>详情查看连接

博文 来自： [star_xiu的博客](#)

大数据分析：利用SVD分解对数据进行降维

阅读数 155

问题描述：写程序利用SVD分解对数据进行降维，并画出降维后各个节点在”新维度“上值的分布。例如对于矩阵 A(...

博文 来自： [崔文韬](#)

基于随机梯度下降的SVD原理分享及Python代码实现

阅读数 939

传统的SVD方法需要将rating分解成用户向量p和物品向量q，通常大型系统中p和q的维数都比较高，例如电商系...

博文 来自： [liumy601的专栏](#)

利用SVD进行图像压缩（附Python代码）

阅读数 692

SVD图像压缩步骤对于RGB图像来说，对应三色层，对每一个色层进行SVD分解，得到U,S,VU, S, VU,S,V矩阵，设利...

博文 来自： [koko_TT的博客](#)

PCA用SVD来实现

阅读数 985

SVD(奇异值分解)是线性代数中一个常见的decomposition；PCA也是dimension reduction领域中的经典之作。初...

博文 来自： [Just for fun的专栏](#)

机器学习笔记(14)——sklearn降维方法举例(RandomProjection,TSVD,t-SNE)

阅读数 1万+

sklearn降维方法举例以datasets.digits数据为例导入相关包import numpy as npimport pandas as pdimport mat...

博文 来自： [icepoint的博客](#)

用截断奇异值分解（Truncated SVD）降维

阅读数 1062

1、SVD的定义2、SVD计算举例3、SVD的一些性质 4、截断SVD截断的SVD将参数计数从u*v减少到t（u+v），如...

博文 来自： [qq_32172681的博客](#)

PCA和SVD降维

阅读数 866

1 问题引入前边几章我们学习了很多机器学习的算法，它们在小规模数据上都很有效，但在实际生活中，我们的数据...

博文 来自： [Howie的博客](#)

SVD应用的经典例子

阅读数 2805

Singular Value Decomposition (SVD) Tutorial

博文 来自： [浴血重生-学习空间](#)

机器学习实战（十二）降维（PCA、SVD）

目录0. 前言1. 主成分分析PCA（Principal Component Analysis） 2. 奇异值分解SVD（Singular Value Decomposi...

博文 来自： [你今天学习了](#)

（十八）通俗易懂理解——SVD降维（协同过滤）

阅读数 432

最近内容看的挺少，但是遇到仍是一大堆不懂的知识点，感觉有很多坑要弥补。这节给自己稍微复习记录一下SVD降...

博文 来自： [qq_36696494的博客](#)

利用Numpy的svd函数实现的PCA为什么是对协方差矩阵进行SVD分解

阅读数 1057

众所周知，PCA是数据分析中经常用到的一种方法，主要用途是对高维数据进行降维，有两大目的：去相关和去冗余...

博文 来自：Blateyang的



2

简单易学的机器学习算法——SVD奇异值分解

阅读



一、SVD奇异值分解的定义

博文 来自：null的专栏

程序员接私活怎样防止做完了不给钱？

阅读数

6

首先跟大家说明一点，我们做 IT 类的外包开发，是非标品开发，所以很有可能在开发过程中会有这样那样的需求修改...

博文 来自：DavidGoGo_f



python json java mysql pycharm android linux json格式



CV_ML_DP

TA的個人主頁>

原創	粉絲	獲贊	評論	訪問
248	319	206	101	43萬+

等級: 博客 6 週排名: 3435

積分: 6917 總排名: 6134

勳章:

關注

私信



分类专栏

- | | | |
|--|------------------|-----|
| | c/c++代码优化 && ... | 1篇 |
| | 跟我一起学机器学习 ... | 95篇 |
| | 跟我一起学算法/代码... | |
| | C++ | 29篇 |
| | Boost | 3篇 |

展开

最新文章

簡單高效刪除數組成員(非穩定性)的代碼

【性能優化】BitMap的使用:(1)創建BitMap (2)插入key到BitMap (3)刪除key到BitMap (4)查詢key是否存在於BitMap中

【鍊錶】雙向單循鍊錶：鍊錶創建、尾部插入節點、節點刪除、指定位置插入節點、向前遍歷、向後遍歷。

【跟我一起學gdb】(11)自動化gdb腳本之file命令+ run命令+ set args命令+ show args命令

【跟我一起學gdb】(10)自動化gdb腳本之list 命令

歸檔



舉報

2019年12月	1篇
2019年2月	2篇
2019年1月	11篇
2018年9月	3篇
2018年6月	10篇
2018年5月	39篇
2018年4月	33篇
2018年3月	52篇
展開	

熱門文章

【機器學習】【線性代數】正交基、標準正交基、正交矩陣，正交變換等數學知識點
閱讀數15573

【機器學習】【邏輯回歸】Logistic函數/Sigmoid函數的詳細公式推導
閱讀數14927

【機器學習】【PCA-1】PCA基本原理和原理推導+ PCA計算步驟講解+ PCA實例展示
閱讀數13976

【安裝工具】【ARM-DS-5】成功破解ARM DS-5 v5.26.0 + 配置高通MDM9026
閱讀數11261

【機器學習】匯總詳解：矩陣的跡以及跡對矩陣求導
閱讀數9993

最新評論

【性能優化】取模運算：x%n，當n...
luoyuegufeng： [reply]u012421852[/reply]x=10 n=6這方法不對啊

【機器學習】【PCA-1】PCA基...
qq_35053889： 你好,博主,你的協方差矩陣C的特徵向量應該是計算錯了吧?你這個算的和我算的 ...

【機器學習】【K-Means】算法...
qq_36627779： 必須懂了大哥

【性能優化】取模運算：x%n，當n...
weixin_31789689： 感謝，很有用。


【機器學習】【SVD-3】SVD降...
rav009： 請問降維降在哪裡了？ 6*4到6*4，哪裡降了？ 這篇文章誤人子弟





QQ客服  kefu@csdn.net


 客服論壇 400-660-0108 
工作時間8:30-22:00


[關於我們](#) [招聘](#) [廣告服務](#) [網站地圖](#)


京ICP備19004658號 經營性網站備案信息
 公安備案號11010502030143
©1999-2020北京創新樂知網絡技術有限公司 網絡110報警服務
北京互聯網违法和不良信息举报中心
中國互聯網舉報中心 家長監護 版權申訴


2





6
















舉報