

|  |
| --- |
|  |
|  |

* HW2
* Yasaman Mirmohammad
* 9431022

Computational Intelligence

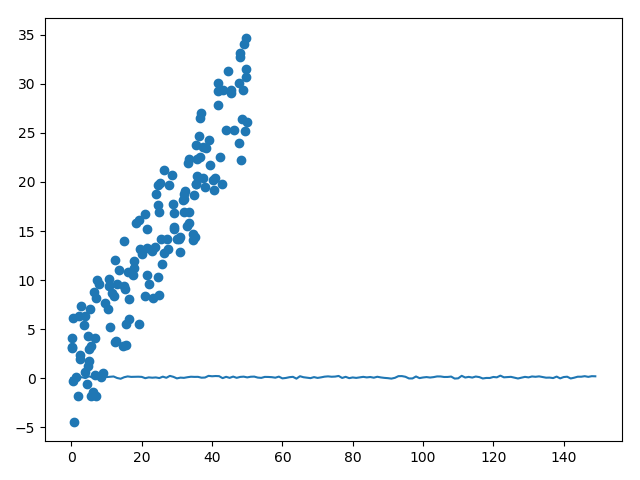
H

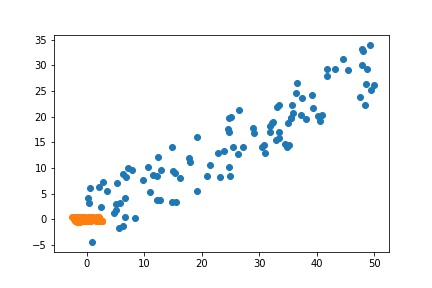
# به نام خدا

# گزارش تمرین دوم هوش محاسباتی

|  |
| --- |
| * نحوه بازنمایی مسئله   دو کلاسES و کروموزوم، مساله را مدل کرده اند.   * نحوه ترکیب کروموزمها با همدیگر و مقدار احتمال ترکیب و تاثیر آن در سرعت همگرایی.   مشاهده شد که هر چه احتمال ترکیب بالاتر برود، سرعت همگرایی بیشتر خواهد شد.  def crossover(self):  num\_crossovers = np.round(len(self.select\_population\_set) \* self.p\_combination)  select\_index\_combination = np.random.randint(0, self.initial\_number \* 3 - 1, int(num\_crossovers))  for i in range(0, len(select\_index\_combination) - 1):  # try:  self.select\_population\_set[select\_index\_combination[i]].gene[0] = \  statistics.mean([self.select\_population\_set[select\_index\_combination[i]].gene[0],  self.select\_population\_set[select\_index\_combination[i + 1]].gene[0]])  self.select\_population\_set[select\_index\_combination[i]].evluate\_update()   self.select\_population\_set[select\_index\_combination[i + 1]].gene[1] = \  statistics.mean([self.select\_population\_set[select\_index\_combination[i]].gene[1],  self.select\_population\_set[select\_index\_combination[i + 1]].gene[1]])  self.select\_population\_set[select\_index\_combination[i + 1]].evluate\_update()  i += 1   * نحوه جهش و نحوه تعیین گام جهش و تاثیر جهش در سرعت همگرایی.   نحوه ی جهش :  chromosome.gene[2] = chromosome.gene[2] \* math.exp(-np.random.normal(0, 1) \* self.sigma) chromosome.gene[0] += N01[0] \* chromosome.gene[2] chromosome.gene[1] += N01[1] \* chromosome.gene[2] norm = ((chromosome.gene[0]) \*\* 2 + (chromosome.gene[1]) \*\* 2) \*\* 0.5 chromosome.gene[0] /= norm chromosome.gene[1] /= norm self.sigma= np.random.normal(0,0.05) ;  با استفاده از یک تابع گاوسی ، ژن ها را جهش میدهیم و نتیجه ی به دست آمده را نرمال سازی میکنیم.   * نحوه انتخاب والدین و بازماندگان.   def choose\_parent(self):  for i in range(1, 4):  select\_index = np.random.randint(0, self.initial\_number - 1, self.initial\_number)   for index in select\_index:  self.select\_population\_set.append(self.initial\_population\_set[index])  def choose\_new\_generation(self):  # *Todo  """  Use one of the discussed methods in class.  Q-tournament is suggested !* ***:return****: Mu selected chromosomes for next cycle  """* select\_chrom\_set = [0, 0, 0, 0]  q\_tornoment = 4   #  chromosome\_set = self.select\_population\_set  chromosome\_set.extend(self.initial\_population\_set)  self.initial\_population\_set.clear()  for i in range(0, self.initial\_number):  select\_index = np.random.randint(0, len(chromosome\_set) - 1 - i, q\_tornoment)  j = 0  for index in select\_index:  select\_chrom\_set[j] = chromosome\_set[index]  j += 1   sorted(select\_chrom\_set, key=lambda x: x.score, reverse=False)  self.initial\_population\_set.append(select\_chrom\_set[0])  chromosome\_set.remove(select\_chrom\_set[0])  self.initial\_population\_set = sorted(self.initial\_population\_set, key=lambda x: x.score, reverse=False)   return   * تعداد جمعیت و تعداد فرزندان )مقادیر مختلف را امتحان کرده و سرعت همگرایی را مشاهده کرده   و نتیجه را ارائه دهید(  تعداد جمعیت و تعداد فرزندان رابطه مستقیمی با سرعت همگرایی دارد چرا که در راستای افزایش تنوع است.   * **شرط خاتمه الگوریتم تکاملی.**   def evolution\_process(self):  self.initial\_population()  i = 0  while (i in range(0, 20)):  for x in self.initial\_population\_set:  print(x.score)    self.select\_parent()  self.mutation()  self.crossover()  self.select\_population()   print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  i += 1  print("best chromosome:", self.initial\_population\_set[0].score,  statistics.stdev(self.initial\_population\_set[0].Z, statistics.mean(self.initial\_population\_set[0].Z)))  print("worst chromosome:", self.initial\_population\_set[self.initial\_number - 1].score)  print("mean score:", statistics.mean(chrom.score for chrom in self.initial\_population\_set))  if i > 100:  break  return self.initial\_population\_set[0]  اگر به پایان تنوع یا شایستگی برسیم الگوریتم خاتمه یافته است.  **توضیحات و نتایج:**  **PCAیک روش تحلیل داده هاست که درآن، گویی ،محورهای مختصات چرخانده میشوند تا ما از زاویه ی دیگری به توزیع داده هایمان نگاه کنیم.**  PCA،تبدیلی در [فضای برداری](https://fa.wikipedia.org/wiki/%D9%81%D8%B6%D8%A7%DB%8C_%D8%A8%D8%B1%D8%AF%D8%A7%D8%B1%DB%8C" \o "فضای برداری) است، که بیشتر برای [کاهش ابعاد](https://fa.wikipedia.org/wiki/%DA%A9%D8%A7%D9%87%D8%B4_%D8%A7%D8%A8%D8%B9%D8%A7%D8%AF" \o "کاهش ابعاد) [مجموعه داده‌ها](https://fa.wikipedia.org/w/index.php?title=%D9%85%D8%AC%D9%85%D9%88%D8%B9%D9%87_%D8%AF%D8%A7%D8%AF%D9%87%E2%80%8C%D9%87%D8%A7&action=edit&redlink=1) مورد استفاده قرار می‌گیرد.  ]تحلیل مولفه‌های اصلی در سال ۱۹۰۱ توسط [کارل پیرسون](https://fa.wikipedia.org/wiki/%DA%A9%D8%A7%D8%B1%D9%84_%D9%BE%DB%8C%D8%B1%D8%B3%D9%88%D9%86" \o "کارل پیرسون) [[۱]](https://fa.wikipedia.org/wiki/%D8%AA%D8%AD%D9%84%DB%8C%D9%84_%D9%85%D8%A4%D9%84%D9%81%D9%87%E2%80%8C%D9%87%D8%A7%DB%8C_%D8%A7%D8%B5%D9%84%DB%8C#cite_note-1) ارائه شد. این تحلیل شامل [تجزیه مقدارهای ویژهٔ](https://fa.wikipedia.org/wiki/%D8%AA%D8%AC%D8%B2%DB%8C%D9%87_%D9%85%D9%82%D8%AF%D8%A7%D8%B1%D9%87%D8%A7%DB%8C_%D9%88%DB%8C%DA%98%D9%87" \o "تجزیه مقدارهای ویژه) [ماتریس کواریانس](https://fa.wikipedia.org/wiki/%D9%85%D8%A7%D8%AA%D8%B1%DB%8C%D8%B3_%DA%A9%D9%88%D8%A7%D8%B1%DB%8C%D8%A7%D9%86%D8%B3) می‌باشد.[  تعریف:یک [تبدیل خطی](https://fa.wikipedia.org/wiki/%D8%AA%D8%A8%D8%AF%DB%8C%D9%84_%D8%AE%D8%B7%DB%8C" \o "تبدیل خطی) [متعامد](https://fa.wikipedia.org/w/index.php?title=%D9%85%D8%AA%D8%B9%D8%A7%D9%85%D8%AF&action=edit&redlink=1) است که داده را به [دستگاه مختصات](https://fa.wikipedia.org/wiki/%D8%AF%D8%B3%D8%AA%DA%AF%D8%A7%D9%87_%D9%85%D8%AE%D8%AA%D8%B5%D8%A7%D8%AA" \o "دستگاه مختصات) جدید می‌برد به طوری که بزرگترین واریانس داده بر روی اولین محور مختصات، دومین بزرگترین واریانس بر روی دومین محور مختصات قرار می‌گیرد و همین طور برای بقیه. تحلیل مولفه‌های اصلی می‌تواند برای کاهش ابعاد داده مورد استفاده قرار بگیرد، به این ترتیب مولفه‌هایی از مجموعه داده را که بیشترین تاثیر در واریانس را دارند حفظ می‌کند   برای [ماتریس داده](https://fa.wikipedia.org/wiki/%D9%85%D8%A7%D8%AA%D8%B1%DB%8C%D8%B3_%D8%AF%D8%A7%D8%AF%D9%87" \o "ماتریس داده)x، با [میانگین تجربی](https://fa.wikipedia.org/w/index.php?title=%D9%85%DB%8C%D8%A7%D9%86%DA%AF%DB%8C%D9%86_%D8%AA%D8%AC%D8%B1%D8%A8%DB%8C&action=edit&redlink=1" \o "میانگین تجربی (صفحه وجود ندارد)) صفر، که هر سطر یک مجموعه مشاهده و هر ستون داده‌های مربوط به یک شاخصه است، تحلیل مولفه‌های اصلی به صورت زیر تعریف می‌شود:  به طوری که :  YT=XTW=V⅀  به طوری که V⅀WT، [تجزیه مقدارهای منفرد](https://fa.wikipedia.org/wiki/%D8%AA%D8%AC%D8%B2%DB%8C%D9%87_%D9%85%D9%82%D8%AF%D8%A7%D8%B1%D9%87%D8%A7%DB%8C_%D9%85%D9%86%D9%81%D8%B1%D8%AF) ماتریس X Tمی‌باشد.  محدودیت ها:  استفاده از تحلیل مولفه‌های اصلی منوط به فرض هایی است که در نظر گرفته می‌شود. از جمله:   * فرض خطی بودن * ما فرض می کنیم مجموعه داده [ترکیب خطی](https://fa.wikipedia.org/wiki/%D8%AA%D8%B1%DA%A9%DB%8C%D8%A8_%D8%AE%D8%B7%DB%8C" \o "ترکیب خطی) پایه‌هایی خاص است. * فرض بر این که میانگین و کواریانس از نظر احتمالاتی قابل اتکا هستند. * فرض بر این که واریانس شاخصه اصلی داده است.   The covariance between two features x\_j and x\_k on the population level can be calculated via the following equation:  https://miro.medium.com/max/200/1*zsJQIL5ZnHRxxqxz0OxntQ.png |
|  |
| μ\_j and μ\_k are the sample means of features j and k.  Want to insert a picture from your files or add a shape, text box, or table? You got it! On the Insert tab of the ribbon, just tap the option you need.  خروجی:  داده ها: |

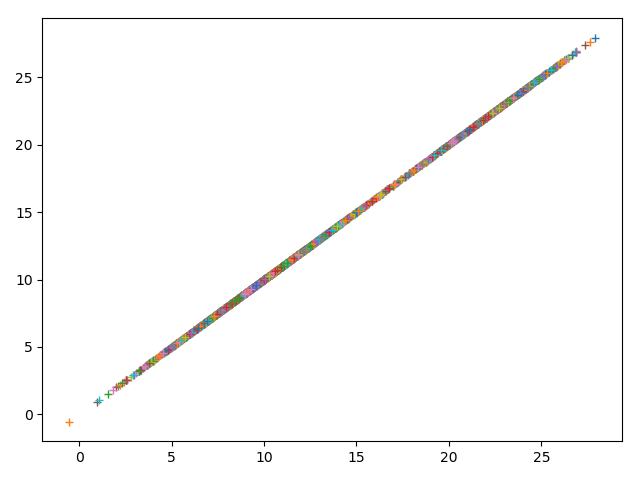
به این شکل مشاهده می شود که داده ها روی یک بعد تصویر شده اند ( محور افقی):





نقاط z تبدیل یافته(نارنجی)

خط pca: در محور داده ها:



بدین شکل مشاهده شد که کاهش بعد انجام شده است.