Неотрицательная матричная факторизация (NMF)

Неотрицательная матричная факторизация - это современный алгоритм извлечения признаков. NMF полезен, когда есть много атрибутов, и атрибуты неоднозначны или имеют слабую предсказуемость. Комбинируя атрибуты, NMF может создавать новые значимые функции, описывающие исследуемый объект. Каждая функция, созданная NMF, представляет собой линейную комбинацию исходного набора атрибутов. У каждого объекта есть набор коэффициентов, которые являются мерой веса каждого атрибута объекта. Есть отдельныйкоэффициент для каждого числового атрибута и для каждого отдельного значения каждого категориального атрибута. Все коэффициенты неотрицательны

Факторизация матрицы

Факторизация неотрицательной матрицы использует методы многомерного анализа и линейной алгебры. Он разлагает данные в виде матрицы M на произведение двух матриц W и H более низкого ранга Подматрица W содержит базис NMF; подматрица H содержит соответствующие коэффициенты (веса).

Алгоритм итеративно модифицирует значения W и H так , что их произведение приближалось к . То есть оптимизируется (минимизируется) расстояние d между X и матричным произведением WH. Наиболее широко используемая функция расстояния - это квадрат нормы Фробениуса, который является очевидным расширением евклидовой нормы на матрицы:

$$d_{Fro}(X,Y) = \frac{1}{2}\left\|X-Y\right\|_{Fro}^2 = \frac{1}{2}\sum_{i,j}(X_{ij}-Y_{ij})^2 \quad (1)$$

При реализации алгоритма NMF используют регуляризованную целевую функцию:

$$d_{Fro}(X,WH)+\alpha\rho\left\|W\right\|_1+\alpha\rho\left\|H\right\|_1+\frac{\alpha(1-\rho)}{2}\left\|W\right\|_{Fro}^2+\frac{\alpha(1-\rho)}{2}\left\|H\right\|_{Fro}^2 \quad (2)$$

где ρ параметр смешивания регуляризации, α константа регуляризации, $\left\|.\right\|_1$ поэлементная норма. Целевая функция (2) минимизируется с попеременной минимизацией W и H.

Алгоритм NMF должен быть инициализирован начальным числом, чтобы указать начальную точку для итераций. Из-за большой размерности пространства обработки и отсутствия алгоритма глобальной мини-

мизации соответствующая инициализация может иметь решающее значение для получения значимых результатов. Часто используют случайное начальное число, которое инициализирует значения W и H на основе равномерного распределения. Этот подход хорошо работает в большинстве случаев.

Неотрицательная матричная факторизация (NMF) может использоваться в качестве этапа предварительной обработки для уменьшения размерности при классификации, регрессии, кластеризации и других задачах интеллектуального анализа данных. Оценка модели NMF производит проекции данных в новое пространство функций. Величина проекции показывает, насколько сильно запись соответствует объекту. Алгоритм NMF сохраняет большую часть структуры исходных данных и гарантирует, что и базис, и веса неотрицательны. Алгоритм завершается, когда ошибка аппроксимации сходится или достигается заданное количество итераций.

Пример реализации алгоритма NMF **на** *python*

Для рачетов использовались данный о Российских регионах. В *python* реализации исходным было $3D$ пространство, регионы размечены на три класса.

В sklearn.decomposition.NMF мы задаем следующие параметры n.components = 2 количество координат для коллектора (размерность выходного редуцируемого пространства), init='random' неотрицательные случайная матрица, масштабируемая с помощью:$\sqrt{X_{mean}/\text{n.components}}$, random.state=0 -начальное значение генератора случайных чисел для воспроизводимости результатов остальные параметры принимаются по умолчанию

Код *python* :

```
  `` python
from sklearn.decomposition import NMF
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import scipy.stats as st
import matplotlib.pyplot as plt
from pandas import Series, DataFrame
import pandas as pd

result =pd.read_table('dan04.txt',sep='\s+')
print(result)
```

```
y=result['P18']

X =DataFrame(result, columns=['P10','P11','P14'])
y=np.array(y)
X=np.array (X)

SUMX=X[:,0] +X[:,1] + X[:,2]
X[:,0] =X[:,0] /SUMX
X[:,1] =X[:,1] /SUMX
X[:,2] =X[:,2] /SUMX

#описательные статистики
print('Описательные статистики предикторов')
for num in range(3) :
  print('Номер переменной: %.0f' %num)
  print('число элентов массива =',len(X[:,num]))
  print('среднее значение =',np.mean(X[:,num]))
  print('стандартное отклонение =',np.std(X[:,num]))
  print('мин мах =',np.min(X[:,num]), np.max(X[:,num]))

#построение исходного графика 3 D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
xs = X[:,0]
ys = X[:,1]
zs =X[:,2]
ax.scatter(xs, ys, zs,c=y, cmap=plt.cm.nipy_spectral,
edgecolor='k')
ax.set_title("Исходное пространство предикторов")
ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
plt.show()

# Преобразование данных , уменьшающее размерность до 2
model = NMF(n_components=2, init='random',
random_state=0 )
```

```
X_transformed = model.fit_transform(X)
H = model.components_
DimX=X_transformed.shape
print(DimX)

y = np.choose(y, [1, 2, 0]).astype(np.float)
plt.clf()
plt.cla()
plt.scatter(X_transformed[:, 0], X_transformed[:, 1],
c=y, cmap=plt.cm.nipy_spectral, edgecolor='k',s=120)
plt.title("Снижение размерности методом NMF")
plt.xlabel("NMF1")
plt.ylabel("NMF2")
plt.show()
```

‘‘‘

Графические результаты работы скрипта python:

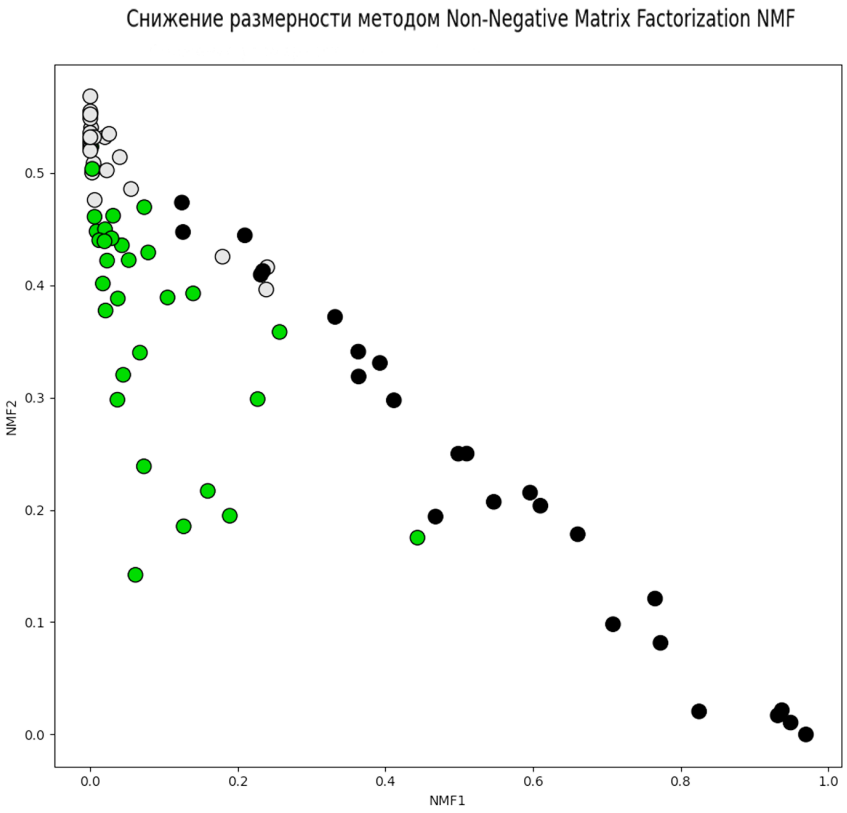

Рис. 1: Регионы в редуцированном 2D пространстве NMF при bandwidth=0.5