CSCI-B 565

Homework 1

John Koo

Problem 1

(Exercise 2.7 from the text)

Daily temperature is likely to display more temporal autocorrelation than daily rainfall. Daily rainfall is zero-inflated.

Problem 2

(Exercise 2.8 from the text)

Asymmetric discrete - word counts are counts, which are discrete, and since the vocabulary size is much larger than the number of documents, the data are zero-inflated.

Problem 3

(Exercise 2.12 from the text)

- The goal of data mining is often to discover the underlying function or system, and noise obfuscates this. Therefore, we often wish to eliminate noise (which is sometimes just the same thing as discovering the underlying function/system). However, noise can be useful in training robust algorithms, characterizing uncertainty, or discovering some systematic problem. Outliers can bias our analyses but should be noted since they can be real phenomena and not just noise
- Noise can result in outliers. For example, if the data are $X_i \overset{iid}{\sim} \mathcal{N}(0, 1)$, then we can have some $X_j > 10$ with nonzero probability, even if that probability is small.
- Noise does not always result in outliers. If our model is some mean μ with some noise/error $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, we can get deviations from μ that are small.
- Outliers are not always due to noise. They could be due to a misspecified model.
- Yes, given noise that are distributed on the real line, we can end up with some high noise value with nonzero probability, even if it is a rare occurrence.

Problem 4

Sample space $\Omega = \{H, T\}$.

This is a discrete sample space. There is a countable number of elements in it.

Problem 5

$$\Omega = \left\{ \begin{array}{cccc} (1,1), & (1,2), & \ldots, & (1,6), \\ & (2,2), & \ldots, & (2,6), \\ & & \vdots & \\ & & & (6,6) \end{array} \right\}$$

There are $\frac{7\times 6}{2} = 21$ elements in Ω. Each element $(x, y) \in \Omega$ has probability $\frac{1}{18}$ unless $x = y$ in which case it has probability $\frac{1}{36}$.

$$A = \left\{ \begin{array}{cccccc} (1,3), & (2,3), & (3,3), & (4,3), & (5,3), & (6,3), \\ (1,4), & (1,5), & (1,6), & (2,4), & (2,5), & (2,6), \\ (4,4), & (4,5), & (4,6), & (5,5), & (5,6), & (6,6) \end{array} \right\}$$

15 elements in A has probability $1/18$ while 3 have probability $1/36$, so $P(A) = 15/18 + 3/36 = 11/12$.

$B = \{(1,2), (2,3), (3,4), (4,5), (5,6)\}$, and every element of B has probability $1/36$, so $P(B) = 5/36$.

$P(A) \times P(B) = \frac{55}{432}$.

$A \cap B = \{(2,3), (3,4), (4,5), (5,6)\}$, so $P(A \cap B) = \frac{2}{9} \neq \frac{55}{432} \implies A$ and B are not independent.

Problem 6

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 5 \end{bmatrix}$$

The characteristic equation is:

$$\begin{aligned} &0 = |M - \lambda I| \\ &= (1-\lambda)((2-\lambda)(5-\lambda)-9) - (5-\lambda-3) + (3-2+\lambda) \\ &= -\lambda^3 + 8\lambda^2 - 6\lambda \end{aligned}$$

We can see that there is one zero eigenvalue and two nonzero eigenvalues, so $rank(M) = 2$.

Let the first eigenvalue be 0. $\lambda_1 = 0$.

The characteristic equation simplifies to:

$$\begin{aligned} &\lambda^2 - 8\lambda + 6 = 0 \\ &\implies \lambda_{2,3} = \frac{8 \pm \sqrt{64-24}}{2} = 4 \pm \sqrt{10} \end{aligned}$$

So the eigenvalues are $(0, 4+\sqrt{10}, 4-\sqrt{10})$.

We can solve for the eigenvectors (v_1, v_2, v_3) by solving $Mv_i = \lambda_i v_i \implies (M - \lambda_i I)v_i = 0$ via gaussian elimination and plugging in λ_i.

Notation: Let $v_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$.

To solve for the first eigenvector v_1:

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 0 \\ 1 & 3 & 5 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

So we can set z_1 to anything arbitrary. Let $z_1 = 0$. Then we can see that $y_1 = -2$ by the second line, and plugging those into the top line yields $x_1 = 1$. So $v_1 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$.

To solve for the second eigenvector v_2:

$$\begin{bmatrix} -3-\sqrt{10} & 1 & 1 & 0 \\ 1 & -2-\sqrt{10} & 3 & 0 \\ 1 & 3 & 1-\sqrt{10} & 0 \end{bmatrix} \to \begin{bmatrix} -3-\sqrt{10} & 1 & 1 & 0 \\ 1 & -2-\sqrt{10} & 3 & 0 \\ 0 & 5+\sqrt{10} & -2-\sqrt{10} & 0 \end{bmatrix}$$

Again, z_2 is arbitrary, so we can set $z_2 = 1$. Plugging this into the third line yields $y_2 = \frac{2+\sqrt{10}}{5+\sqrt{10}} = \frac{\sqrt{10}}{5}$.

Plugging y_2 and z_2 into the top line yields

$$(-3-\sqrt{10})x_2 = -\frac{\sqrt{10}+5}{5}$$
$$\implies x_1 = -1 + \frac{2\sqrt{10}}{5}$$

So $v_2 = \begin{bmatrix} -1+\frac{2\sqrt{10}}{5} \\ \frac{\sqrt{10}}{5} \\ 1 \end{bmatrix}$

Since eigenvectors are orthogonal, the third eigenvector is just the conjugate of the second.

$$v_3 = \begin{bmatrix} -1-\frac{2\sqrt{10}}{5} \\ -\frac{\sqrt{10}}{5} \\ 1 \end{bmatrix}$$

I will skip normalizing the eigenvectors here.

Problem 7

For brevity, I will omit some of the EDA code from the original notebook.

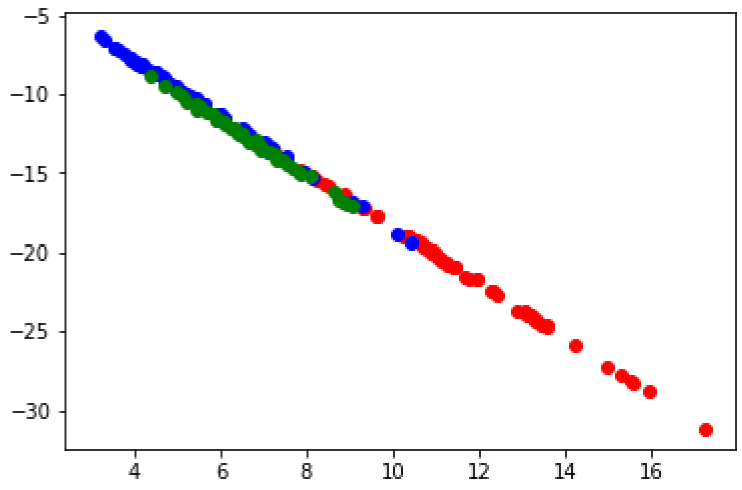
In [1]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

wine = pd.read_csv('wine.csv')
label = wine.pop('Class')
wine_scaled = StandardScaler().fit_transform(wine)
pca = PCA(n_components=2).fit(wine_scaled)

wine_np = np.asarray(wine)
```

```
/home/johnkoo/venv/lib/python3.6/site-packages/sklearn/preprocessing/
data.py:625: DataConversionWarning: Data with input dtype int64, floa
t64 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/home/johnkoo/venv/lib/python3.6/site-packages/sklearn/base.py:462: D
ataConversionWarning: Data with input dtype int64, float64 were all c
onverted to float64 by StandardScaler.
  return self.fit(X, **fit_params).transform(X)
```

In [2]:
```
pc_scaled = np.matmul(wine_np, pca.components_.T) / pca.singular_valu
es_
```

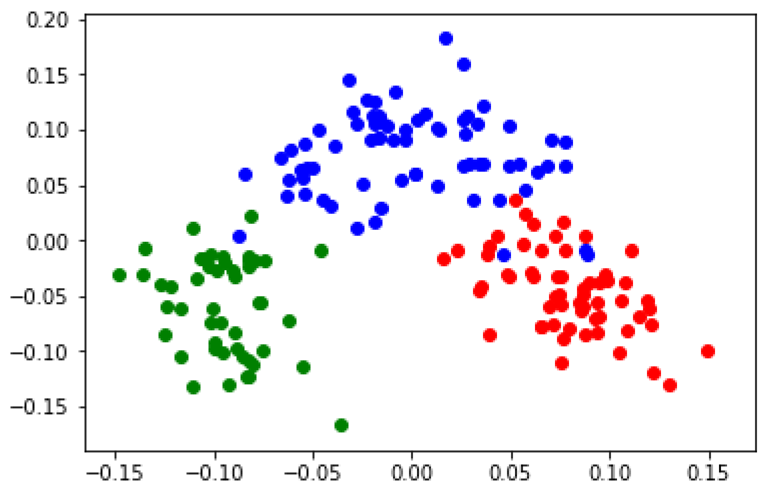
In [3]:
```
plt.scatter(pc_scaled[label == 1, 0], pc_scaled[label == 1, 1], c='re
d')
plt.scatter(pc_scaled[label == 2, 0], pc_scaled[label == 2, 1], c='bl
ue')
plt.scatter(pc_scaled[label == 3, 0], pc_scaled[label == 3, 1], c='gr
een')
```

Out[3]:
```
<matplotlib.collections.PathCollection at 0x7f64ab953b38>
```



In [4]:
```
# if we want to use manually scaled data
pc_scaled = np.matmul(
    (wine_np - np.mean(wine_np, axis=0)) / np.std(wine_np, axis=0),
    pca.components_.T) / pca.singular_values_

plt.scatter(pc_scaled[label == 1, 0], pc_scaled[label == 1, 1], c='re
d')
plt.scatter(pc_scaled[label == 2, 0], pc_scaled[label == 2, 1], c='bl
ue')
plt.scatter(pc_scaled[label == 3, 0], pc_scaled[label == 3, 1], c='gr
een')
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x7f64a98f7a20>
```



Problem 8

In [5]:
```
appointments = pd.read_csv('KaggleV2-May-2016.csv')

appointments.head()
```

Out[5]:

	PatientId	AppointmentID	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood
0	2.987250e+13	5642903	F	2016-04-29T18:38:08Z	2016-04-29T00:00:00Z	62	JARDIM DA PENHA
1	5.589978e+14	5642503	M	2016-04-29T16:08:27Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA
2	4.262962e+12	5642549	F	2016-04-29T16:19:04Z	2016-04-29T00:00:00Z	62	MATA DA PRAIA
3	8.679512e+11	5642828	F	2016-04-29T17:29:31Z	2016-04-29T00:00:00Z	8	PONTAL DE CAMBURI
4	8.841186e+12	5642494	F	2016-04-29T16:07:23Z	2016-04-29T00:00:00Z	56	JARDIM DA PENHA

In [6]:
```
print(appointments.shape)

print(appointments.columns)

appointments.describe()
```

```
(110527, 14)
Index(['PatientId', 'AppointmentID', 'Gender', 'ScheduledDay',
       'AppointmentDay', 'Age', 'Neighbourhood', 'Scholarship', 'Hipe
rtension',
       'Diabetes', 'Alcoholism', 'Handcap', 'SMS_received', 'No-sho
w'],
      dtype='object')
```

Out[6]:

	PatientId	AppointmentID	Age	Scholarship	Hipertension	Diabet
count	1.105270e+05	1.105270e+05	110527.000000	110527.000000	110527.000000	110527.0000
mean	1.474963e+14	5.675305e+06	37.088874	0.098266	0.197246	0.0718
std	2.560949e+14	7.129575e+04	23.110205	0.297675	0.397921	0.2582
min	3.921784e+04	5.030230e+06	-1.000000	0.000000	0.000000	0.0000
25%	4.172614e+12	5.640286e+06	18.000000	0.000000	0.000000	0.0000
50%	3.173184e+13	5.680573e+06	37.000000	0.000000	0.000000	0.0000
75%	9.439172e+13	5.725524e+06	55.000000	0.000000	0.000000	0.0000
max	9.999816e+14	5.790484e+06	115.000000	1.000000	1.000000	1.0000

The data have 110,527 rows and 14 columns. Each column except for the last corresponds to an attribute, but two of the columns, `PatientID` and `AppointmentID` , have no real meaning in analysis or modeling, so there are really only 11 attributes that matter.

We can see that `Age` is a numeric variable (age should be continuous, but in this case because of the precision of these values, they are discrete). `Gender` , `Scholarship` , `Hypertension` , `Diabetes` , `Alcoholism` , and `SMS_received` are binary $\{0, 1\}$ variables. `Neighborhood` is a categorical variable. `ScheduledDay` and `AppointmentDay` are numeric variables, and since they are timestamps, they are interval values. At first glance, it appears that there are no missing values, but we have some -1 values in the `Age` column, which probably indicate that missing values there were coded as such.

For a classification model, it makes sense to look at all 11 attributes (excluding `PatientID` and `AppointmentID`). One transformed attribute that we may want to consider is the time interval between scheduled and appointment days.