written2

September 30, 2024

1 STP598 Machine Learning & Deep Learning

1.1 Written Assignment 2

1.1.1 Due 11:59pm Wednesday Oct. 9, 2024 on Canvas

1.1.2 name, id

1.2 Question 1 (Negative Bionomial Regression)

In class we talked about Poisson regression for counting data. Notice that the variance is equal to the mean $\mathrm{Var}(Y) = \mathrm{E}(Y)$. However in certain circumstances the observed variance is bigger than the mean, known as *overdispersion* and Poisson becomes an inappropriate model. To address this issue, alternative models such as **negative binomial** model and **zero-inflated** model are proposed.

In negative binomial regression, the counts Y are modeled using the negative binomial distribution, which usually characterizes the distribution of random number of successes before r failures with success probability p:

$$Y \sim \mathrm{NB}(r,p), \qquad \Pr(Y = y|r,p) = \binom{y+r-1}{y} p^y (1-p)^r$$

with the mean $\mathrm{E}(Y) = \mu = \frac{pr}{1-p}$ and the variance $\mathrm{Var}(Y) = \mu(1 + \mu/r)$. Reparametrize it in terms of r and μ and we actually get negative binomial represented as **Gamma-Poisson** mixture:

$$Y|\lambda \sim \mathrm{Pois}(\lambda)\lambda|r,\mu \sim \Gamma(r, \mu/r)$$

with the density rewritten as

$$\Pr(Y = y|r,\mu) = \frac{\Gamma(y+r)}{\Gamma(y+1)\Gamma(r)} \left(\frac{\mu}{r+\mu}\right)^y \left(\frac{r}{r+\mu}\right)^r$$

Now in generalized linear model of negative binomial regression given covariates X, what is the link function? Can you write down the log-likelihood function in terms of r and regression parameter β? Briefly explain how you will fit the model if r is relaxed to be a real number?

1.3 Question 2

The Iris dataset represents 3 kind of Iris flowers (Setosa, Versicolour and Virginica) with 4 attributes: sepal length, sepal width, petal length and petal width.

Principal Component Analysis (PCA) applied to this data identifies the combination of attributes (principal components, or directions in the feature space) that account for the most variance in the data. We can plot the different samples on the first 2 principal components.

We first load the data set.

[1]:

```
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data
y = iris.target
target_names = iris.target_names
```

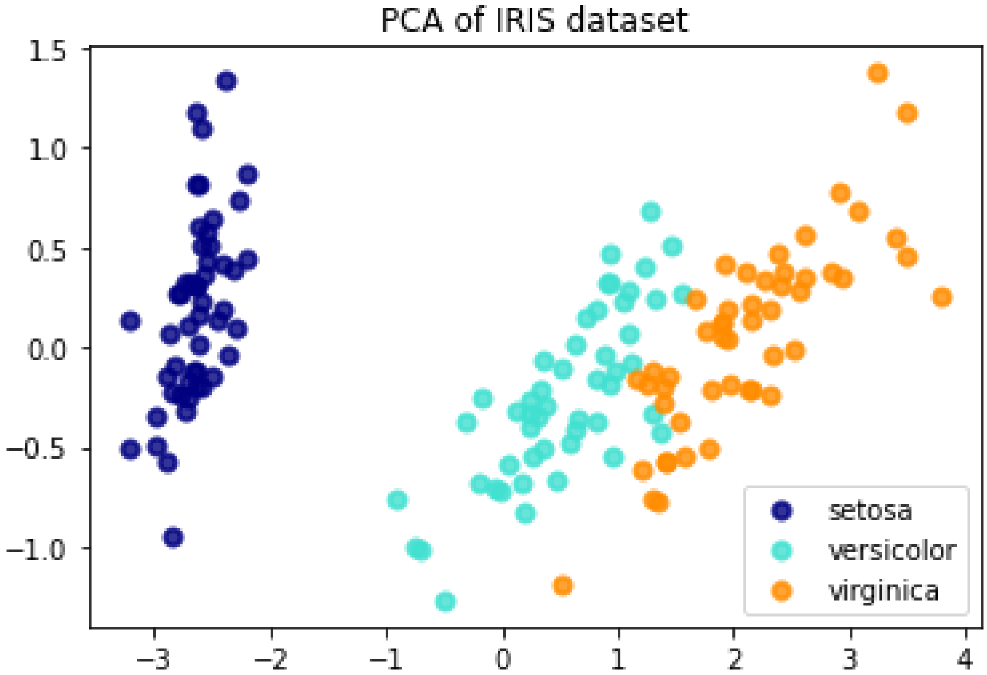
Now we run PCA on this data set and plot the samples on the first 2 principal directions.

[2]:

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
%matplotlib inline

pca = PCA(n_components=2)
X_r = pca.fit(X).transform(X)
plt.figure()
colors = ['navy', 'turquoise', 'darkorange']
lw = 2

for color, i, target_name in zip(colors, [0, 1, 2], target_names):
    plt.scatter(X_r[y == i, 0], X_r[y == i, 1], color=color, alpha=.8, lw=lw,
                label=target_name)
plt.legend(loc='best', shadow=False, scatterpoints=1)
plt.title('PCA of IRIS dataset')
plt.show()
```



- Now make the similar plots using LDA (`sklearn.discriminant_analysis.LinearDiscriminantAnalysis`). Set *n_components=2* in the LDA function.

Save such 2d latent outputs by LDA as `Xr2`. Now use `X_r2` and `y` to fit the following models:

1. LDA and logistic regression (`sklearn.linear_model.LogisticRegression`). Plot their decision boundaries on the same picture (Please use different colors or line styles to differentiate them).
2. QDA (`sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis`) and LDA with quadratic terms. Plot their decision boundaries on the same picture (Please use different colors or line styles to differentiate them).