DSO530 Statistical Learning Methods

Lecture 4a: Classification IV

Dr. Xin Tong
Department of Data Sciences and Operations
Marshall School of Business
University of Southern California

Review question 1

Recall that a linear classifier means that its decision boundary is linear. Please show that logistic regression gives us linear classifiers. Concretely, show the classifier

$$1\left(\frac{e^{3+9x_1+5x_2}}{1+e^{3+9x_1+5x_2}} > 0.8\right)$$

has a linear decision boundary.

Review question 2

What is the linear discriminant analysis model?

Review question 3

Is the following claim TRUE or FALSE: “As I collect more and more observations, I can usually achieve (close to) perfect classification.”

Review question 4

Is the following claim TRUE or FALSE: “For logistic regression, when I increase the decision threshold from 0.5 to 0.6, the type I error will increase as well”

K-nearest neighbors (KNN)

- 'KNN': to predict class label for an observation $X = x$, the K training observations that are closest to x are identified. Then x is assigned to the class to which the plurality of these observations belong
- Why did we say "plurality" instead of *majority*?
- Can predict the class label for any $x \in R^p$ (including the training observations) in this way
- Special cases: $K = 1$ and $K = n$
- KNN is a very different approach to classification. Can you name some differences from logistic regression and LDA?
- On training dataset, which choice of K gives the smallest error? Will you choose this K?
- In general, as K increases, how will training error change? What about test error?
- Does KNN have linear decision boundaries?

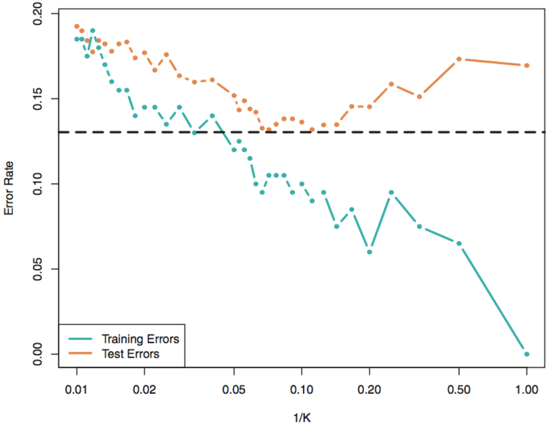


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

Figure 1: Training and test errors vs. $1/K$

```python
import numpy as np
import pandas as pd
url="https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
# Assign colum names to the dataset
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']
# Read dataset to pandas dataframe
dataset = pd.read_csv(url, names=names); dataset.head()
```

```
##    sepal-length  sepal-width  petal-length  petal-width        Class
## 0           5.1          3.5           1.4          0.2  Iris-setosa
## 1           4.9          3.0           1.4          0.2  Iris-setosa
## 2           4.7          3.2           1.3          0.2  Iris-setosa
## 3           4.6          3.1           1.5          0.2  Iris-setosa
## 4           5.0          3.6           1.4          0.2  Iris-setosa
```

- This is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

```python
np.unique(dataset["Class"])
```

```
## array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```python
dataset.info()
```

```
## <class 'pandas.core.frame.DataFrame'>
## RangeIndex: 150 entries, 0 to 149
## Data columns (total 5 columns):
## sepal-length    150 non-null float64
## sepal-width     150 non-null float64
## petal-length    150 non-null float64
## petal-width     150 non-null float64
## Class           150 non-null object
## dtypes: float64(4), object(1)
## memory usage: 6.0+ KB
```

```
dataset.describe()
```

```
##        sepal-length  sepal-width  petal-length  petal-width
## count    150.000000   150.000000    150.000000   150.000000
## mean       5.843333     3.054000      3.758667     1.198667
## std        0.828066     0.433594      1.764420     0.763161
## min        4.300000     2.000000      1.000000     0.100000
## 25%        5.100000     2.800000      1.600000     0.300000
## 50%        5.800000     3.000000      4.350000     1.300000
## 75%        6.400000     3.300000      5.100000     1.800000
## max        7.900000     4.400000      6.900000     2.500000
```

- The four predictors are on the same scale
- In the slides, we will not use feature rescaling. After you go home, please try the feature rescaling step and see if the result becomes better.

```python
from sklearn.model_selection import train_test_split
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values
X_train, X_test, y_train, y_test=\
  train_test_split(X, y, test_size=0.20, random_state=5, stratify=y)

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)
```

```
## KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
##                      metric_params=None, n_jobs=None, n_neighbors=5, p=2,
##                      weights='uniform')
```

- There are many default parameters here. We just pay attention to `weights`.

```
y_pred = classifier.predict(X_test)
np.mean(y_pred != y_test)
```

```
## 0.03333333333333333
```

- After you go home, try practice different settings, such as `random_state=100`, `n_neighbors = 3`, `n_neighbors = 7`

Understanding different errors

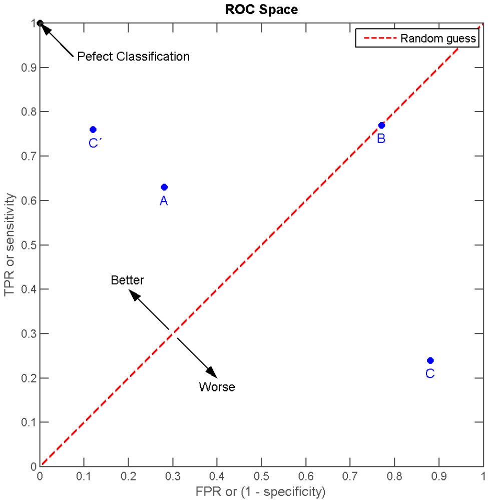
		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

Figure 2

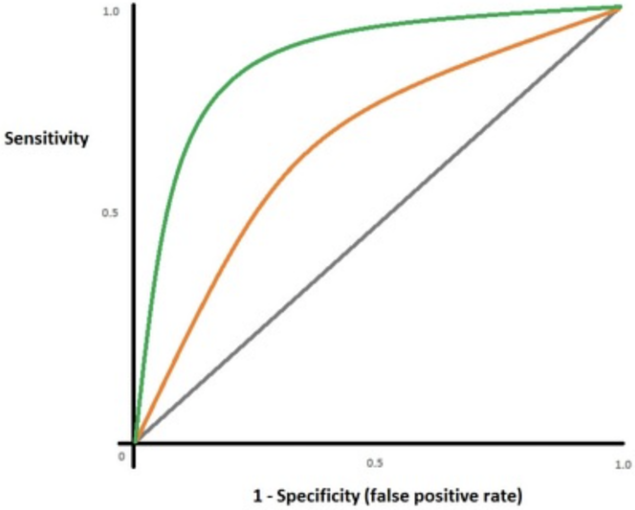
Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

Figure 3

Receiver operating characteristic (ROC) space



Receiver operating characteristic (ROC) curves



- Which curve is better? One way to judge: area under the curve (AUC)

Data science vs. statistics

- Optional and for fun only
- https://www.youtube.com/watch?v=uHGlCi9jOWY&feature=youtu.be&fbclid=IwAR1tTAIEhgUrHk815BlMj0PxC8XYRLz62pA_V6C__xSWXk3ZAF8uoj9I0JQ
- In some sense, the first half of DSO 530 is more of statistics, and the second half is more data science (machine learning)