TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II

Deadline 17/10/2022 (Monday) 23:59 via Fenix as PDF

I. Pen-and-paper [13v]

Four positive observations, $\left\{\binom{A}{0},\binom{B}{1},\binom{A}{1},\binom{A}{0}\right\}$, and four negative observations, $\left\{\binom{B}{0},\binom{B}{0},\binom{A}{1},\binom{B}{1}\right\}$, were collected. Consider the problem of classifying observations as positive or negative.

1) [4v] Compute the recall of a distance-weighted kNN with $k=5$ and distance $d(\mathbf{x}_1,\mathbf{x}_2)=Hamming(\mathbf{x}_1,\mathbf{x}_2)+\frac{1}{2}$ using leave-one-out evaluation schema (i.e., when classifying one observation, use all remaining ones).

	z_i	5NN	weighted mode	$\hat{z}_i$
$\mathbf{x}_1$	P	$\{\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}$	$\left\{\frac{8}{3}P,2N\right\}$	P (TP)
$\mathbf{x}_2$	P	$\{\mathbf{x}_3,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7,\mathbf{x}_8\}$	$\left\{\frac{2}{3}P,\frac{10}{3}N\right\}$	N (FN)
$\mathbf{x}_3$	P	$\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_4,\mathbf{x}_7,\mathbf{x}_8\}$	$\left\{2P,\frac{8}{3}N\right\}$	N (FN)
$\mathbf{x}_4$	P	$\{\mathbf{x}_1,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}$	$\left\{\frac{8}{3}P,2N\right\}$	P (TP)
$\mathbf{x}_5$	N	$\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_6,\mathbf{x}_8\}$	$\left\{2P,\frac{8}{3}N\right\}$	N (TN)
$\mathbf{x}_6$	N	$\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_5,\mathbf{x}_8\}$	$\left\{2P,\frac{8}{3}N\right\}$	N (TN)
$\mathbf{x}_7$	N	$\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_8\}$	$\left\{\frac{10}{3}P,\frac{2}{3}N\right\}$	P (FP)
$\mathbf{x}_8$	N	$\{\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}$	$\left\{\frac{8}{3}P,2N\right\}$	P (FP)

$$Recall = TP/(TP+FN) = 2/(2+2) = 1/2$$

An additional positive observation was acquired, $\binom{B}{0}$, and a third variable y_3 was independently monitored, yielding estimates $y_3|P=\{1.2, 0.8, 0.5, 0.9, 0.8\}$ and $y_3|N=\{1, 0.9, 1.2, 0.8\}$.

2) [4v] Considering the nine training observations, learn a Bayesian classifier assuming: i) y_1 and y_2 are dependent, ii) $\{y_1,y_2\}$ and $\{y_3\}$ variable sets are independent and equally important, and ii) y_3 is normally distributed. Show all parameters.

$$p(z|\mathbf{x})=\frac{p(\mathbf{x}|z)\times p(z)}{p(\mathbf{x})}$$

$$p(z)\text{: } \textbf{priors}\text{: } p(P)=\frac{5}{9},\ \ p(N)=\frac{4}{9}$$

$$p(\mathbf{x}|z)\text{: } \textbf{PMFs}\text{: } p(A,0|P)=\frac{2}{5},\ \ p(A,1|P)=\frac{1}{5},\ \ p(B,0|P)=\frac{1}{5},\ \ p(B,1|P)=\frac{1}{5}$$

$$p(A,0|N)=0,\ \ p(A,1|N)=\frac{1}{4},\ \ p(B,0|N)=\frac{2}{4},\ \ p(B,1|N)=\frac{1}{4}$$

$$\textbf{PDFs}\text{: } N(u_3=0.84,\sigma_3=0.25\mid P),\ \ N(u_3=0.975,\sigma_3=0.17\mid N)$$

$p(\mathbf{x})$ is <u>optional</u> for classification purposes

$$p(A,0)=\frac{2}{9},\ \ p(A,1)=\frac{2}{9},\ \ p(B,0)=\frac{3}{9},\ \ p(B,1)=\frac{2}{9},\ \ N(u_3=0.9,\sigma_3=0.218)$$

Aprendizagem 2022/23

Homework II

Deadline 17/10/2022 (Monday) 23:59 via Fenix as PDF

Considering three testing observations, $\left\{\left(\begin{pmatrix}A\\1\\0.8\end{pmatrix}, \text{Positive}\right), \left(\begin{pmatrix}B\\1\\1\end{pmatrix}, \text{Positive}\right), \left(\begin{pmatrix}B\\0\\0.9\end{pmatrix}, \text{Negative}\right)\right\}$.

3) [3v] Under a MAP assumption, compute $P(\text{Positive}|\mathbf{x})$ of each testing observation.

Let us compute the $p(\text{z}|\mathbf{x})$ for each observation:

$$\mathbf{x}_1: p(A,1,0.8|P) = p(A,1|P)p(0.8|P) = \frac{1}{5} \times 1.574 = 0.315,\ \ p(A,1,0.8|N) = \frac{1}{4} \times 1.38 = 0.345$$

$$p(P|A,1,0.8) = 0.315 \times \frac{5}{9} \times k,\ \ p(N|A,1,0.8) = 0.345 \times \frac{4}{9} \times k$$

$$\text{normalization: } p(P|A,1,0.8) = \frac{p(\mathbf{x}|P)p(P)}{p(\mathbf{x}|P)p(P) + p(\mathbf{x}|N)p(N)} = \frac{p(P|\mathbf{x})}{p(P|\mathbf{x}) + p(N|\mathbf{x})} = 0.533$$

$$\mathbf{x}_2: p(B,1,1|P) = p(B,1|P)p(1|P) = \frac{1}{5} \times 1.3 = 0.26,\ \ p(B,1,1|N) = \frac{1}{4} \times 2.32 = 0.58,$$

$$p(P|B,1,1) = 0.26 \times \frac{5}{9} \times k,\ \ p(N|B,1,1) = 0.58 \times \frac{4}{9} \times k$$

$$\text{after normalization: } p(P|B,1,1) = \ 0.36$$

$$\mathbf{x}_3: p(B,0,0.9|P) = p(B,0|P)p(0.9|P) = \frac{1}{5} \times 1.55 = 0.31,\ \ p(B,0,0.9|N) = \frac{2}{4} \times 2.13 = 1.065,$$

$$p(P|B,0,0.9) = 0.31 \times \frac{5}{9} \times k,\ \ p(N|B,0,0.9) = 1.065 \times \frac{4}{9} \times k$$

$$\text{after normalization: } p(P|B,0,0.9) = 0.27$$

4) [2v] Given a binary class variable, the default decision threshold of $\theta = 0.5$,

$$f(\mathbf{x}|\theta) = \begin{cases} \text{Positive} & P(\text{Positive}|\mathbf{x}) > \theta \\ \text{Negative} & \text{otherwise} \end{cases}$$

can be adjusted. Which decision threshold – 0.3, 0.5 or 0.7 – optimizes testing accuracy?

At 0.7: accuracy = 1/3
At 0.5: accuracy = 2/3
At 0.3: accuracy = 1 ← target threshold

II. Programming and critical analysis [7v]

Considering the `pd_speech.arff` dataset available at the course webpage's homework tab.

1) [3v] Using `sklearn`, considering a 10-fold stratified cross validation (`random=0`), plot the cumulative confusion matrices of kNN (uniform weights, $k = 5$, Euclidean distance) and Naïve Bayes (Gaussian assumption). Use all remaining classifier parameters as default.

```
kNN confusion matrix:
                        predicted
                      Healthy  Parkinson
observed  Healthy      49       143
          Parkinson    64       500

NB confusion matrix:
                        predicted
                      Healthy  Parkinson
observed  Healthy      65       127
          Parkinson    68       496
```

```
kNN accuracies: 0.73 ± 0.04
 [0.7, 0.71, 0.78, 0.79, 0.72, 0.71, 0.75, 0.71, 0.77, 0.63]
NB accuracies: 0.74 ± 0.08
 [0.61, 0.74, 0.8, 0.88, 0.76, 0.66, 0.84, 0.75, 0.76, 0.63]
```

```
from sklearn import metrics
from sklearn.model_selection import StratifiedKFold
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB

knn_accs, nb_accs = [], []
knn_cm, nb_cm = np.array([[0,0],[0,0]]), np.array([[0,0],[0,0]])
folds = StratifiedKFold(n_splits=10, random_state=0)
knn_predictor = KNeighborsClassifier(n_neighbors=5)
nb_predictor = GaussianNB()

# iterate per fold
for train_k, test_k in folds.split(X, y):
    X_train, X_test = X.iloc[train_k], X.iloc[test_k]
    y_train, y_test = y.iloc[train_k], y.iloc[test_k]

    # train and assess
    knn_predictor.fit(X_train, y_train)
    nb_predictor.fit(X_train, y_train)
    knn_pred, nb_pred = knn_predictor.predict(X_test), nb_predictor.predict(X_test)

    knn_cm = knn_cm + np.array(metrics.confusion_matrix(y_test, knn_pred, labels=['0','1']))
    nb_cm = nb_cm + np.array(metrics.confusion_matrix(y_test, nb_pred, labels=['0','1']))
    knn_accs.append(round(metrics.accuracy_score(y_test, knn_pred),2))
    nb_accs.append(round(metrics.accuracy_score(y_test, nb_pred),2))

print("kNN confusion matrix:\n",pd.DataFrame(knn_cm, index=['Healthy', 'Parkinson'], columns=['Healthy', 'Parkinson']))
print("NB confusion matrix:\n",pd.DataFrame(nb_cm, index=['Healthy', 'Parkinson'], columns=['Healthy', 'Parkinson']))
print("kNN accuracies:",round(np.mean(knn_accs),2),"±",round(np.std(knn_accs),2),"\n",knn_accs)
print("NB accuracies:",round(np.mean(nb_accs),2),"±",round(np.std(nb_accs),2),"\n",nb_accs)
```

2) [2v] Using `scipy`, test the hypothesis "kNN is statistically superior to Naïve Bayes regarding accuracy", asserting whether is true.

```
res = stats.ttest_rel(knn_accs, nb_accs, alternative='greater')
```

For the specifications in the statement, p-value=0.91. One cannot reject the null hypothesis at common significance levels (e.g., $\alpha = 0.1$), and thus we cannot assert the given hypothesis as true. Note that we should refrain from stating that the given hypothesis is false or rejected in the absence of additional statistical tests.

3) [2v] Enumerate three reasons underlying the observed performance differences between kNN and Naïve Bayes.

Possibilities in favor of NB, i.e. against kNN: **1**) lack of data normalization, strongly affecting distances; **2**) locality of decisions (small k) in face of a large population; **3**) high data dimensionality with potential uninformative variables, further affecting the adequacy of distances between observations; **4**) suboptimality of kNN hyperparameterizations (e.g., weighting scheme, distance choice)…

Possibilities in favor of kNN, i.e. against NB were also considered when grading, including: **1**) variable dependencies (inadequacy of independence assumption); **2**) variables not normally distributed (inadequacy of Gaussian assumption); **3**) probability estimates from a limited number of observations (e.g., inadequate estimates, null probabilities); **4**) imbalanced class creating biases in MAP estimates via priors; …

END