Title

discrim knn — kth-nearest-neighbor discriminant analysis

Description

`discrim knn` performs kth-nearest-neighbor discriminant analysis. A wide selection of similarity and dissimilarity measures is available.

kth-nearest neighbor must retain the training data and search through the data for the k nearest observations each time a classification or prediction is performed. Consequently for large datasets, kth-nearest neighbor is slow and uses a lot of memory.

See [MV] **discrim** for other discrimination commands.

Quick start

kth-nearest-neighbor discriminant analysis of `v1`, `v2`, `v3`, and `v4` for groups defined by `catvar` and $k = 5$

```
discrim knn v1 v2 v3 v4, k(5) group(catvar)
```

Same as above, but use prior probabilities proportional to group size

```
discrim knn v1 v2 v3 v4, k(5) group(catvar) priors(proportional)
```

Display only the leave-one-out classification table

```
discrim knn v1 v2 v3 v4, k(5) group(catvar) lootable notable
```

Use absolute-value distance

```
discrim knn v1 v2 v3 v4, k(5) group(catvar) measure(absolute)
```

Assume `v1` and `v2` are factor variables, and use the Dice similarity coefficient

```
discrim knn ibn.v1 ibn.v2, k(5) group(catvar) measure(dice)
```

Menu

Statistics > Multivariate analysis > Discriminant analysis > Kth-nearest neighbor (KNN)

Syntax

```
discrim knn varlist [if] [in] [weight], group(groupvar) k(#) [options]
```

options	Description
Model	
*group(*groupvar*)	variable specifying the groups
*k(#)	number of nearest neighbors
priors(*priors*)	group prior probabilities
ties(*ties*)	how ties in classification are to be handled
Measure	
measure(*measure*)	similarity or dissimilarity measure; default is measure(L2)
s2d(standard)	convert similarity to dissimilarity: $d(ij) = \sqrt{s(ii) + s(jj) - 2s(ij)}$, the default
s2d(oneminus)	convert similarity to dissimilarity: $d(ij) = 1 - s(ij)$
mahalanobis	Mahalanobis transform continuous data before computing dissimilarities
Reporting	
notable	suppress resubstitution classification table
lootable	display leave-one-out classification table

priors	Description
equal	equal prior probabilities; the default
proportional	group-size-proportional prior probabilities
matname	row or column vector containing the group prior probabilities
matrix_exp	matrix expression providing a row or column vector of the group prior probabilities

ties	Description
missing	ties in group classification produce missing values; the default
random	ties in group classification are broken randomly
first	ties in group classification are set to the first tied group
nearest	ties in group classification are assigned based on the closest observation, or missing if this still results in a tie

*group() and k() are required.

varlist may contain factor variables; see [U] 11.4.3 Factor variables.

collect and statsby are allowed; see [U] 11.1.10 Prefix commands.

fweights are allowed; see [U] 11.1.6 weight.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

`group(`*groupvar*`)` is required and specifies the name of the grouping variable. *groupvar* must be a numeric variable.

`k(`*#*`)` is required and specifies the number of nearest neighbors on which to base computations. In the event of ties, the next largest value of `k()` is selected. Suppose that `k(3)` is selected. For a given observation, one must go out a distance d to find three nearest neighbors, but if, say, there are five data points all within distance d, then the computation will be based on all five nearest points.

`priors(`*priors*`)` specifies the prior probabilities for group membership. The following *priors* are allowed:

`priors(equal)` specifies equal prior probabilities. This is the default.

`priors(proportional)` specifies group-size-proportional prior probabilities.

`priors(`*matname*`)` specifies a row or column vector containing the group prior probabilities.

`priors(`*matrix_exp*`)` specifies a matrix expression providing a row or column vector of the group prior probabilities.

`ties(`*ties*`)` specifies how ties in group classification will be handled. The following *ties* are allowed:

`ties(missing)` specifies that ties in group classification produce missing values. This is the default.

`ties(random)` specifies that ties in group classification are broken randomly.

`ties(first)` specifies that ties in group classification are set to the first tied group.

`ties(nearest)` specifies that ties in group classification are assigned based on the closest observation, or missing if this still results in a tie.

Measure

`measure(`*measure*`)` specifies the similarity or dissimilarity measure. The default is `measure(L2)`; all measures in [MV] ***measure_option*** are supported except for `measure(Gower)`.

`s2d(standard | oneminus)` specifies how similarities are converted into dissimilarities.

The available `s2d()` options, `standard` and `oneminus`, are defined as

$$\texttt{standard} \qquad d(ij) = \sqrt{s(ii) + s(jj) - 2s(ij)} = \sqrt{2\{1 - s(ij)\}}$$

$$\texttt{oneminus} \qquad d(ij) = 1 - s(ij)$$

`s2d(standard)` is the default.

`mahalanobis` specifies performing a Mahalanobis transformation on continuous data before computing dissimilarities. The data are transformed via the Cholesky decomposition of the within-group covariance matrix, and then the selected dissimilarity measure is performed on the transformed data. If the `L2` (Euclidean) dissimilarity is chosen, this is the Mahalanobis distance. If the within-group covariance matrix does not have sufficient rank, an error is returned.

Reporting

notable suppresses the computation and display of the resubstitution classification table.

lootable displays the leave-one-out classification table.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction
A first example
Mahalanobis transformation
Binary data

Introduction

kth-nearest-neighbor (KNN) discriminant analysis dates at least as far back as Fix and Hodges (1951). An introductory treatment is available in Rencher and Christensen (2012). More advanced treatments are in Hastie, Tibshirani, and Friedman (2009) and McLachlan (2004).

KNN is a nonparametric discrimination method based on the k nearest neighbors of each observation. KNN can deal with binary data via one of the binary measures; see [MV] ***measure_option***.

A first example

What distinguishes kth-nearest-neighbor analysis from other methods of discriminant analysis is its ability to distinguish irregular-shaped groups, including groups with multiple modes. We create a dataset with unusual boundaries that lends itself to KNN analysis and graphical interpretation.

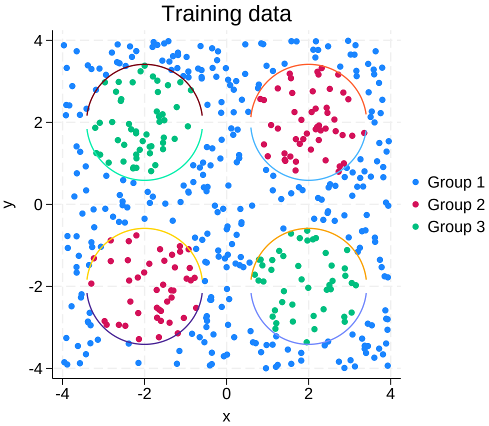
▷ Example 1

We create a two-dimensional dataset on the plane with x and y values in $[-4, 4]$. In each quadrant we consider points within a circle with a square root of two radii, centered around the points $(2, 2)$, $(-2, 2)$, $(-2, -2)$, and $(2, -2)$. We set the group value to 1 to start and then replace it in the circles. In the first and third circles we set the group value to 2, and in the second and fourth circles we set the group value to 3. Outside the circles, the group value remains 1.

```
. set seed 98712321
. set obs 500
Number of observations (_N) was 0, now 500.
. generate x = 8*runiform() - 4
. generate y = 8*runiform() - 4
. generate group = 1
. replace group = 2 if (y+2)^2 + (x+2)^2 <= 2
(45 real changes made)
. replace group = 2 if (y-2)^2 + (x-2)^2 <= 2
(50 real changes made)
. replace group = 3 if (y+2)^2 + (x-2)^2 <= 2
(45 real changes made)
. replace group = 3 if (y-2)^2 + (x+2)^2 <= 2
(51 real changes made)
```

Next we define some local macros for function plots of the circles. This makes it easier to graph the boundary circles on top of the data. We set the graph option aspectratio(1) to force the aspect ratio to be 1; otherwise, the circles might appear to be ovals.

```
. local rp : di %12.10f 2+sqrt(2)
. local rm : di %12.10f 2-sqrt(2)
. local functionplot
> (function y =  sqrt(2-(x+2)^2) - 2, lpat(solid) range(-‘rp’ -‘rm’))
> (function y = -sqrt(2-(x+2)^2) - 2, lpat(solid) range(-‘rp’ -‘rm’))
> (function y =  sqrt(2-(x-2)^2) + 2, lpat(solid) range(-‘rm’  ‘rp’))
> (function y = -sqrt(2-(x-2)^2) + 2, lpat(solid) range(-‘rm’  ‘rp’))
> (function y =  sqrt(2-(x+2)^2) + 2, lpat(solid) range(-‘rp’ -‘rm’))
> (function y = -sqrt(2-(x+2)^2) + 2, lpat(solid) range(-‘rp’ -‘rm’))
> (function y =  sqrt(2-(x-2)^2) - 2, lpat(solid) range( ‘rm’  ‘rp’))
> (function y = -sqrt(2-(x-2)^2) - 2, lpat(solid) range( ‘rm’  ‘rp’))
. local graphopts
> aspectratio(1) legend(order(1 "Group 1" 2 "Group 2" 3 "Group 3"))
. twoway (scatter y x if group==1)
>        (scatter y x if group==2)
>        (scatter y x if group==3)
>        ‘functionplot’ , ‘graphopts’ name(original, replace)
>        title("Training data")
```



We perform three discriminant analyses on these data for comparison. We use linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and KNN. The results from logistic discriminant analysis are similar to those of LDA and are not included. With all three models, we use proportional probabilities, `priors(proportional)`. The probability of landing in a given group is proportional to the geometric area of that group; they are certainly not equal. Rather than doing geometric calculations for the prior probabilities, we use `priors(proportional)` to approximate this. We suppress the standard classification table with `notable`. Instead we look at the `lootable`, that is, leave-one-out (LOO) table, where the observation in question is omitted and its result is predicted from the rest of the data. Likewise, we predict the LOO classification (`looclass`). With KNN we get to choose a `measure()`; here we want the straight line distance between the points. This is the default, Euclidean distance, so we do not have to specify `measure()`.

We choose $k = 7$ for this run with 500 observations. See *Methods and formulas* for more information on choosing k.

```
. discrim lda x y, group(group) notable lootable priors(proportional)
Linear discriminant analysis
Leave-one-out classification summary

  ┌─────────┐
  │ Key     │
  ├─────────┤
  │ Number  │
  │ Percent │
  └─────────┘

             │ Classified
True group   │      1        2        3 │   Total
─────────────┼──────────────────────────┼────────
           1 │    309        0        0 │     309
             │ 100.00     0.00     0.00 │  100.00
             │                          │
           2 │     95        0        0 │      95
             │ 100.00     0.00     0.00 │  100.00
             │                          │
           3 │     96        0        0 │      96
             │ 100.00     0.00     0.00 │  100.00
─────────────┼──────────────────────────┼────────
       Total │    500        0        0 │     500
             │ 100.00     0.00     0.00 │  100.00
             │                          │
      Priors │ 0.6180   0.1900   0.1920 │
```

LDA classifies all observations into group one, the group with the highest prior probability.

```
. discrim qda x y, group(group) notable lootable priors(proportional)
Quadratic discriminant analysis
Leave-one-out classification summary

  ┌─────────┐
  │ Key     │
  ├─────────┤
  │ Number  │
  │ Percent │
  └─────────┘

             │ Classified
True group   │      1        2        3 │   Total
─────────────┼──────────────────────────┼────────
           1 │    258       31       20 │     309
             │  83.50    10.03     6.47 │  100.00
             │                          │
           2 │     57       38        0 │      95
             │  60.00    40.00     0.00 │  100.00
             │                          │
           3 │     77        0       19 │      96
             │  80.21     0.00    19.79 │  100.00
─────────────┼──────────────────────────┼────────
       Total │    392       69       39 │     500
             │  78.40    13.80     7.80 │  100.00
             │                          │
      Priors │ 0.6180   0.1900   0.1920 │
```

QDA has 185 $(31 + 20 + 57 + 77)$ misclassified observations of 500, but it correctly classifies 38 of the 95 observations from group 2 and 19 of the 96 observations from group 3.

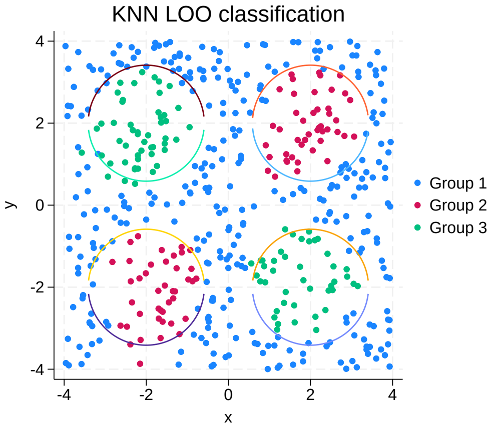
```
. discrim knn x y, group(group) k(7) notable lootable priors(proportional)
Kth-nearest-neighbor discriminant analysis
Leave-one-out classification summary

      +---------+
      | Key     |
      |---------|
      | Number  |
      | Percent |
      +---------+

             | Classified
  True group |       1        2        3 |  Total
-------------+---------------------------+--------
           1 |     299        4        6 |    309
             |   96.76     1.29     1.94 | 100.00
             |                           |
           2 |      13       82        0 |     95
             |   13.68    86.32     0.00 | 100.00
             |                           |
           3 |      10        0       86 |     96
             |   10.42     0.00    89.58 | 100.00
-------------+---------------------------+--------
       Total |     322       86       92 |    500
             |   64.40    17.20    18.40 | 100.00
             |                           |
      Priors |  0.6180   0.1900   0.1920 |
```

In contrast to the other two models, KNN has only 33 $(4+6+13+10)$ misclassified observations. We can see how points are classified by KNN by looking at the following graph.

```
. predict cknn, looclass
. twoway (scatter y x if cknn==1 )
>         (scatter y x if cknn ==2)
>         (scatter y x if cknn ==3)
>         ‘functionplot’, ‘graphopts’ name(knn, replace)
>         title("KNN LOO classification")
```



KNN has some resolution of the circles and correctly classifies most of the points. Its misclassified observations are near the boundaries of the circles, where nearest points fall on both sides of the boundary line.

◁

Mahalanobis transformation

The Mahalanobis transformation eliminates the correlation between the variables and standardizes the variance of each variable, as demonstrated in example 2 of [MV] **discrim lda**. When the Mahalanobis transformation is used in conjunction with Euclidean distance, it is called Mahalanobis distance. The Mahalanobis transformation may be applied when any continuous measure is chosen, not just `measure(Euclidean)`. See [MV] ***measure_option*** for a description of the available measures.

▷ Example 2

We will reproduce an example from Rencher and Christensen (2012, 290–292) that uses the Mahalanobis distance. Rencher and Christensen present data collected by G. R. Bryce and R. M. Barker of Brigham Young University as part of a preliminary study of a possible link between football helmet design and neck injuries. Six head dimensions were measured for each subject. Thirty subjects were sampled in each of three groups: high school football players (group 1), college football players (group 2), and nonfootball players (group 3). The six variables are `wdim`, head width at its widest dimension; `circum`, head circumference; `fbeye`, front-to-back measurement at eye level; `eyehd`, eye to top of head measurement; `earhd`, ear to top of head measurement; and `jaw`, jaw width.

These measurements will not have the same ranges. For example, the head circumference should be much larger than eye to top of head measurement. Mahalanobis distance is used to standardize the measurements.

```
. use https://www.stata-press.com/data/r18/head, clear
(Table 8.3. Head measurements, Rencher and Christensen (2012))

. discrim knn wdim-jaw, k(5) group(group) mahalanobis

Kth-nearest-neighbor discriminant analysis
Resubstitution classification summary

    +---------+
    | Key     |
    |---------|
    | Number  |
    | Percent |
    +---------+

                 | Classified                                          |
    True group   | High school      College    Nonplayer  Unclassified |
    -------------+-----------------------------------------------------+
     High school |          26            0            1             3 |
                 |       86.67         0.00         3.33         10.00 |
                 |                                                     |
         College |           1           19            9             1 |
                 |        3.33        63.33        30.00          3.33 |
                 |                                                     |
       Nonplayer |           1            4           22             3 |
                 |        3.33        13.33        73.33         10.00 |
    -------------+-----------------------------------------------------+
           Total |          28           23           32             7 |
                 |       31.11        25.56        35.56          7.78 |
                 |                                                     |
          Priors |      0.3333       0.3333       0.3333               |

                 | Classified
    True group   |      Total
    -------------+------------
     High school |          30
                 |      100.00
                 |
         College |          30
                 |      100.00
                 |
       Nonplayer |          30
                 |      100.00
    -------------+------------
           Total |          90
                 |      100.00
                 |
          Priors |
```

A subset of this result is in Rencher and Christensen (2012, 331–332). Of the 90 original observations, 16 were misclassified and 7 observations were unclassified. Rencher and Christensen also state the error rate for this example is 0.193. We use `estat errorrate` to get the error rate.

```
. estat errorrate

Error rate estimated by error count

                 | group
                 | High school      College    Nonplayer |        Total
    -------------+---------------------------------------+-------------
      Error rate |  .037037037   .344827586   .185185185 |   .189016603
    -------------+---------------------------------------+-------------
          Priors |  .333333333   .333333333   .333333333 |
Note: 7 observations were not classified and are not included in the table.
```

Our error rate of 0.189 does not match that of Rencher and Christensen. Why is this? Rencher and Christensen calculates the error rate as the number of misclassified observations over the total number of observations classified. This is 16/83 ≈ 0.193. We use the standard error-rate definition that takes into account the prior probabilities. From the high school group, there is one misclassified observation of 27 total observations classified from this group, so its error rate is (1/27) ≈ 0.037, and its contribution to the total is (1/27)(1/3). Likewise, the error rates for the college and nonplayer group are (10/29) ≈ 0.345 and (5/27) ≈ 0.185 respectively, with contributions of (10/29)(1/3) and (5/27)(1/3). Adding all contributions, we get the displayed error rate of 0.189. See

Methods and formulas of [MV] **discrim estat** for details.

The unclassified observations are those that resulted in ties. We can force ties to be classified by changing the `ties()` option. The default is `ties(missing)`, which says that ties are to be classified as missing values. Here we choose `ties(nearest)`, which breaks the tie by classifying to the group of the nearest tied observation.

```
. discrim knn wdim-jaw, k(5) group(group) mahalanobis ties(nearest)
Kth-nearest-neighbor discriminant analysis
Resubstitution classification summary

    +---------+
    | Key     |
    |---------|
    | Number  |
    | Percent |
    +---------+

              | Classified
   True group | High school    College   Nonplayer |      Total
 -------------+--------------------------------------+-----------
  High school |          28          0           2 |         30
              |       93.33       0.00        6.67 |     100.00
              |                                    |
      College |           1         20           9 |         30
              |        3.33      66.67       30.00 |     100.00
              |                                    |
    Nonplayer |           1          4          25 |         30
              |        3.33      13.33       83.33 |     100.00
 -------------+--------------------------------------+-----------
        Total |          30         24          36 |         90
              |       33.33      26.67       40.00 |     100.00
              |                                    |
       Priors |      0.3333     0.3333      0.3333 |
```

Compare this with example 1 in [MV] **candisc**, example 3 in [MV] **discrim estat**, and example 2 of [MV] **discrim logistic**.

◁

Binary data

In addition to the measures for continuous data, a variety of binary measures are available for KNN. Binary data can be created from any categorical dataset by using `xi`; see [R] **xi**.

▷ Example 3

You have invited some scientist friends over for dinner, including Mr. Mushroom (see vignette below), a real "fun guy". Mr. Mushroom is not only a researcher in mycology who enjoys studying mushrooms but also an enthusiastic mushroom gourmand who likes nothing better than to combine his interests in classification and cookery. His current research is identification of poisonous mushrooms from photographs. From the photographs, he can identify the shape of a mushroom's cap, the cap's surface, the cap's color, the population of mushrooms, and, with some careful attention to detail in the surrounding area, the habitat.

> William Alphonso Murrill (1867–1957) was a famous mycologist, taxonomist, and writer from the New York Botanical Gardens and was nicknamed "Mr. Mushroom". Although we borrowed his nickname, Mr. Mushroom and the events portrayed in this example are entirely fictitious. William Murrill's many scientific accomplishments include the 1916 book *Edible and Poisonous Mushrooms*.

Knowing your friend, you imagine that he will insist on bringing a mushroom dish to be unveiled and served at dinnertime—perhaps his experimental subjects. Although you know that he is a careful scientist and a gifted cook, you are stalked with worries about poisoning your other guests.

Late at night you cannot sleep for worrying about poisonous mushrooms, and you decide to do a little research into mushroom classification. You do a Google search online and find mushroom data at http://archive.ics.uci.edu/ml/datasets/Mushroom. For reference, these records are drawn from Lincoff (1981).

This is a large dataset of 8,124 observations on the *Agaricus* and *Lepiota*. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This last class was combined with the poisonous one. Lincoff (1981) clearly states that there is no simple rule for determining the edibility of a mushroom; no rule like "leaflets three, let it be" for Poison Oak and Ivy, a fact that does not seem comforting. Twenty-two attributes are collected, including those that Mr. Mushroom can identify from his photographs.

The mushroom data is a set of 23 variables that describe the cap of the mushroom, whether or not it has bruises, the gills, the veil, the stalk, the ring, the spores, the population, and the habitat. The variables that describe the cap, for example, are `capshape`, `capsurface`, and `capcolor`. The `capshape` variable, for example, has categories bell, conical, convex, flat, knobbed, and sunken. Other variables and categories are similar.

You read in this dataset by using `infile` and make some modifications, attaching notes to this dataset to describe what you did to the original mushroom data. Modifications include dropping categories of the variables of interest that completely determine whether a mushroom is poisonous. The full mushroom data are also available; `webuse mushroom_full` to obtain it.

```
. use https://www.stata-press.com/data/r18/mushroom
(Lincoff (1981) Audubon Guide; http://archive.ics.uci.edu/ml/datasets/Mushroom)

. tabulate habitat poison

           |        poison
   habitat |    edible  poisonous |     Total
-----------+----------------------+----------
   grasses |       752        680 |     1,432 
    leaves |       240        585 |       825 
   meadows |       128         24 |       152 
     paths |       136      1,008 |     1,144 
     urban |        64        224 |       288 
     woods |     1,848      1,268 |     3,116 
-----------+----------------------+----------
     Total |     3,168      3,789 |     6,957 
```

You can see by tabulating two of the variables, `habitat` and `poison`, that in each habitat you have some mushrooms that are poisonous as well as some that are edible. The other descriptive variables of interest produce similar results.

Each variable is a set of unordered categories. Thus, you can treat them as factor variables. Because your goal is to account for all categories, you will apply the factor-variable base operator `ibn.` to the categorical variables. For details, see [U] **11.4.3 Factor variables**.

With KNN you can choose a measure that is suited to these data. You expect data with many zeroes and few ones. A match of two ones is far more significant than two matching zeroes. Looking through the binary similarity measures in [MV] ***measure_option***, you see that the Jaccard binary similarity coefficient reports the proportion of matches of ones when at least one of the observations contains a one, and the Dice binary similarity measure weighs matches of ones twice as heavily as the Jaccard measure. Either suits the situation, and you choose the Dice measure. The conversion from a similarity to a dissimilarity measure will be `s2d(standard)` by default.

The poisonous and edible mushrooms are split about half and half in the original dataset, and in the current subset of these data the ratio is still approximately half and half, so you do not specify priors, obtaining `priors(equal)`, the default.

Because of the size of the dataset and the number of indicator variables created by the factor-variable base operator `ibn.`, KNN analysis is slow. You decide to discriminate based on 2,000 points selected at random, approximately a third of the data.

```
. set seed 12345678
. generate u = runiform()
. sort u
. discrim knn ibn.population ibn.habitat ibn.bruises ibn.capshape
> ibn.capsurface ibn.capcolor in 1/2000, k(15) group(poison) measure(dice)
Kth-nearest-neighbor discriminant analysis
Resubstitution classification summary

    +---------+
    | Key     |
    |---------|
    | Number  |
    | Percent |
    +---------+

                |     Classified
    True poison |    edible  poisonous |     Total
    ------------+----------------------+-----------
         edible |       848         65 |       913
                |     92.88       7.12 |    100.00
                |                      |
      poisonous |        29      1,058 |     1,087
                |      2.67      97.33 |    100.00
    ------------+----------------------+-----------
          Total |       877      1,123 |     2,000
                |     43.85      56.15 |    100.00
                |                      |
         Priors |    0.5000     0.5000 |
```

In some settings, these results would be considered good. Of the original 2,000 mushrooms, you see that only 29 poisonous mushrooms have been misclassified as edible. However, even sporadic classification of a poisonous mushroom as edible is a much bigger problem than classifying an edible mushroom as poisonous. This does not take the cost of misclassification into account. You decide that calling a poisonous mushroom edible is at least 10 times worse than calling an edible mushroom poisonous. In the two-group case, you can easily use the `priors()` option to factor in this cost; see [MV] **discrim** or McLachlan (2004, 9). We set the prior probability of poisonous mushrooms 10 times higher than that of the edible mushrooms.

```
. estat classtable in 1/2000, priors(.09, .91)
Resubstitution classification table

    +---------+
    | Key     |
    |---------|
    | Number  |
    | Percent |
    +---------+

             | Classified
 True poison |    edible  poisonous |     Total
-------------+----------------------+-----------
      edible |       728        185 |       913
             |     79.74      20.26 |    100.00
             |                      |
   poisonous |         0      1,087 |     1,087
             |      0.00     100.00 |    100.00
-------------+----------------------+-----------
       Total |       728      1,272 |     2,000
             |     36.40      63.60 |    100.00
             |                      |
      Priors |    0.0900     0.9100 |
```

These results are reassuring. There are no misclassified poisonous mushrooms, although 185 edible mushrooms of the total 2,000 mushrooms in our model are misclassified.

You now check to see how this subsample of the data performs in predicting the poison status of the rest of the data. This takes a few minutes of computer time, but unlike using `estat classtable` above, the variable predicted will stay with your dataset until you drop it. `tabulate` can be used instead of `estat classtable`.

```
. predict cpoison, classification priors(.09, .91)
. label values cpoison poison
. tabulate poison cpoison
           |   classification
    poison |    edible  poisonous |     Total
-----------+----------------------+----------
    edible |     2,450        718 |     3,168
 poisonous |         0      3,789 |     3,789
-----------+----------------------+----------
     Total |     2,450      4,507 |     6,957
```

This is altogether reassuring. Again, no poisonous mushrooms were misclassified. Perhaps there is no need to worry about dinnertime disasters, even with a fungus among us. You are so relieved that you plan on serving a Jello dessert to cap off the evening—your guests will enjoy a mold to behold. Under the circumstances, you think doing so might just be a “morel” imperative.

◁

Stored results

`discrim knn` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_groups)`	number of groups
`e(k_nn)`	number of nearest neighbors
`e(k)`	number of discriminating variables

Macros

`e(cmd)`	`discrim`
`e(subcmd)`	`knn`
`e(cmdline)`	command as typed
`e(groupvar)`	name of group variable
`e(grouplabels)`	labels for the groups
`e(measure)`	similarity or dissimilarity measure
`e(measure_type)`	`dissimilarity` or `similarity`
`e(measure_binary)`	`binary`, if binary measure specified
`e(s2d)`	`standard` or `oneminus`, if `s2d()` specified
`e(varlist)`	discriminating variables
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(ties)`	how ties are to be handled
`e(mahalanobis)`	`mahalanobis`, if Mahalanobis transform is performed
`e(properties)`	`nob noV`
`e(estat_cmd)`	program used to implement `estat`
`e(predict)`	program used to implement `predict`
`e(marginsnotok)`	predictions disallowed by `margins`

Matrices

`e(groupcounts)`	number of observations for each group
`e(grouppriors)`	prior probabilities for each group
`e(groupvalues)`	numeric value for each group
`e(SSCP_W)`	pooled within-group SSCP matrix
`e(W_eigvals)`	eigenvalues of `e(SSCP_W)`
`e(W_eigvecs)`	eigenvectors of `e(SSCP_W)`
`e(S)`	pooled within-group covariance matrix
`e(Sinv)`	inverse of `e(S)`
`e(sqrtSinv)`	Cholesky (square root) of `e(Sinv)`
`e(community)`	community of neighbors for prediction

Functions

`e(sample)`	marks estimation sample

Methods and formulas

Let g be the number of groups, n_i the number of observations for group i, and q_i the prior probability for group i. Let $\mathbf{x}$ denote an observation measured on p discriminating variables. For consistency with the discriminant analysis literature, $\mathbf{x}$ will be a column vector, though it corresponds to a row in your dataset. Let $f_i(\mathbf{x})$ represent the density function for group i, and let $P(\mathbf{x}|G_i)$ denote the probability of observing $\mathbf{x}$ conditional on belonging to group i. Denote the posterior probability of group i given observation $\mathbf{x}$ as $P(G_i|\mathbf{x})$. With Bayes's theorem, we have

$$P(G_i|\mathbf{x}) = \frac{q_i f_i(\mathbf{x})}{\sum_{j=1}^{g} q_j f_j(\mathbf{x})}$$

Substituting $P(\mathbf{x}|G_i)$ for $f_i(\mathbf{x})$, we have

$$P(G_i|\mathbf{x}) = \frac{q_i P(\mathbf{x}|G_i)}{\sum_{j=1}^{g} q_j P(\mathbf{x}|G_j)}$$

For KNN discrimination, we let k_i be the number of the k nearest neighbors from group i, and the posterior-probability formula becomes

$$P(G_i|\mathbf{x}) = \frac{\dfrac{q_i k_i}{n_i}}{\displaystyle\sum_{j=1}^{g} \frac{q_j k_j}{n_j}}$$

In the event that there are ties among the nearest neighbors, k is increased to accommodate the ties. If five points are all nearest and equidistant from a given $\mathbf{x}$, then an attempt to calculate the three nearest neighbors of $\mathbf{x}$ will actually obtain five nearest neighbors.

Determining the nearest neighbors depends on a dissimilarity or distance calculation. The available dissimilarity measures are described in [MV] ***measure_option***. Continuous and binary measures are available. If a similarity measure is selected, it will be converted to a dissimilarity by either

`standard` $\quad d(ij) = \sqrt{s(ii) + s(jj) - 2s(ij)} = \sqrt{2\{1 - s(ij)\}}$

`oneminus` $\quad d(ij) = 1 - s(ij)$

With any of the continuous measures, a Mahalanobis transformation may be performed before computing the dissimilarities. For details on the Mahalanobis transformation, see *Methods and formulas* of [MV] **discrim lda**. The Mahalanobis transformation with Euclidean distance is called Mahalanobis distance.

Optimal choice of k for KNN is not an exact science. With two groups, k should be chosen as an odd integer to avoid ties. Rencher and Christensen (2012, 331) cites the research of Loftsgaarden and Quesenberry (1965), which suggests that an optimal k is $\sqrt{n_i}$, where n_i is a typical group size. Rencher and Christensen also suggest running with several different values of k and choosing the one that gives the best error rate. McLachlan (2004) cites Enas and Choi (1986), which suggests that when there are two groups of comparable size that k should be chosen approximately between $N^{3/8}$ or $N^{2/8}$, where N is the number of observations.

References

Enas, G. G., and S. C. Choi. 1986. Choice of the smoothing parameter and efficiency of *k*-nearest neighbor classification. *Computers and Mathematics with Applications* 12A: 235–244. https://doi.org/10.1016/0898-1221(86)90076-3.

Fix, E., and J. L. Hodges. 1951. Discriminatory analysis: Nonparametric discrimination, consistency properties. In *Technical Report No. 4, Project No. 21-49-004*. Randolph Field, Texas: Brooks Air Force Base, USAF School of Aviation Medicine.

Hastie, T. J., R. J. Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

Kimbrough, J. W. 2003. The twilight years of William Alphonso Murrill. http://www.mushroomthejournal.com/.

Lincoff, G. H. 1981. *National Audubon Society Field Guide to North American Mushrooms (National Audubon Society Field Guide Series)*. New York: Alfred A. Knopf.

Loftsgaarden, D. O., and C. P. Quesenberry. 1965. A nonparametric estimate of a multivariate density function. *Annals of Mathematical Statistics* 36: 1049–1051. https://doi.org/10.1214/aoms/1177700079.

McLachlan, G. J. 2004. *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley.

Murrill, W. A. 1916. *Edible and Poisonous Mushrooms*. Published by the author.

Rencher, A. C., and W. F. Christensen. 2012. *Methods of Multivariate Analysis*. 3rd ed. Hoboken, NJ: Wiley.

Rose, D. W. 2002. William Alphonso Murrill: The legend of the naturalist. *Mushroom, The Journal of Wild Mushrooming*. http://www.mushroomthejournal.com/.

Smith-Vikos, T. 2003. William Alphonso Murrill (1869–1957). *The New York Botanical Garden*. http://sciweb.nybg.org/science2/hcol/intern/murrill1.asp.

Also see

[MV] **discrim knn postestimation** — Postestimation tools for discrim knn

[MV] **discrim** — Discriminant analysis

[U] **20 Estimation and postestimation commands**

For suggested citations, see the FAQ on citing Stata documentation.