Practical\_Python\_AI\_Projects\_c02

55

Table 2-14. Multi-Period Blending Results

Buy qty Month 0 Month 1 Month 2 Month 3 Month 4

o0 1935.7 0.0 0.0 0.0 0.0

o1 480.7 0.0 274.6 0.0 0.0

o2 192.4 0.0 545.9 0.0 0.0

o3 2835.0 1553.3 0.0 0.0 0.0

o4 293.7 0.0 0.0 136.8 0.0

o5 0.0 966.7 1611.3 0.0 0.0

o6 482.6 1011.5 275.1 1517.9 0.0

o7 0.0 0.0 0.0 1247.9 0.0

o8 0.0 1468.5 2293.1 597.4 0.0

Blend qty Month 0 Month 1 Month 2 Month 3 Month 4

o0 1683.6 117.7 149.4 0.0 2034.4

o1 532.7 0.0 274.6 0.0 919.5

o2 113.3 272.1 269.3 276.6 105.6

o3 1551.3 1465.1 1524.0 0.0 382.6

o4 363.7 0.0 0.0 136.8 392.7

o5 141.0 966.7 1051.8 559.5 0.0

o6 525.6 684.9 601.7 1517.9 1165.2

o7 0.0 25.0 0.0 747.9 0.0

o8 89.0 1468.5 1129.2 1761.3 0.0

(continued)

Chapter 2 Linear Continuous ModeLs

56

Hold qty Month 0 Month 1 Month 2 Month 3 Month 4

o0 15.0 267.2 149.4 0.0 0.0

o1 52.0 0.0 0.0 0.0 0.0

o2 193.0 272.1 0.0 276.6 0.0

o3 152.0 1435.7 1524.0 0.0 0.0

o4 70.0 0.0 0.0 0.0 0.0

o5 141.0 0.0 0.0 559.5 0.0

o6 43.0 0.0 326.6 0.0 0.0

o7 25.0 25.0 0.0 0.0 500.0

o8 89.0 0.0 0.0 1163.9 0.0

Prod qty 5000.0 5000.0 5000.0 5000.0 5000.0

P. Cost $735098.96 $616064.04 $644688.93 $491829.66 $0.00

S. Cost $3900.00 $10000.00 $10000.00 $10000.00 $2500.00

Acid % Month 0 Month 1 Month 2 Month 3 Month 4

a0 13.6 13.3 13.3 14.6 14.6

a1 24.9 24.5 25.2 25.5 23.2

a2 17.8 18.5 17.8 17.8 19.7

a3 3.7 3.7 3.7 3.7 4.1

a4 5.0 5.0 5.0 5.0 5.0

a5 8.8 8.8 8.8 9.7 9.7

a6 26.1 26.1 26.1 23.6 23.6

Total 100.0 100.0 100.0 100.0 100.0

Table 2-14. (continued)

Chapter 2 Linear Continuous ModeLs

57

• Instead of satisfying some demand, we may be asked

maximize profit. In this case, we need to know the price

of the final product, which of course may change at

each month.

• The inventory levels may be stated in terms of each oil

instead of aggregate quantities.

• There may be uncertainty in the fatty acid content of

certain oils.

2.5 Pattern Classification

Classification is currently one of the most successful applications of software

to tasks that were, not so long ago, the privilege of the human intellect.

For instance, software decides if an email is legitimate or spam, whether a

biopsied cell is malignant or benign, and whether the company should offer

you an interview or let your re´sume´ rot in the great bit bucket in the sky.

Let’s look at one of the first effective techniques for the binary

classification of data. The example is contrived because I want to draw

pictures to guide the intuition, but the code we will write is applicable in a

wide variety of cases.

Let’s imagine that we are trying to automate the classification of cells as

malignant or benign based on two measures: the area and the perimeter.

Those features are measured automatically from a picture of the cell under

a microscope. The process starts with a collection of such cells, divided

by an expert into the two groups. These groups form what is known as the

training set for our software. After we have “trained” our software, we will

feed it new data, that has not been seen by an expert, and it will decide in

which group the cell falls. That is, it will classify the cell as malignant or

benign. This process is real and used in laboratories all over the world. The

major simplification I am making here is that many more than two features

are used in practice.

Chapter 2 Linear Continuous ModeLs

58

Let’s consider as an example the cell features plotted in Figure 2-3

with perimeter on the x-axis and radius on the y-axis. We see that the

two classes can be separated by a line. Our task is to discover that line. Of

course, there are a number of valid lines but, as a first attempt, any line

separating the two classes will do.

2.5.1 Constructing a Model

Algebraically, a line is an equation of the form a1 x1 + a2 x2 = b for some

fixed coefficient a1 , a2 , a0 . Or, in dimension n, we call it a hyperplane and it

has an equation of

i

n

i ia x a

=

å =

1

0

Figure 2-3. Cell data and separation hyperplane

Chapter 2 Linear Continuous ModeLs

59

What does it mean for a particular point x to be on one side or the

other of the line? It means that either a1 x1 + a2 x2 < a0 or a1 x1 + a2 x2 > a0 .

These strict inequalities can be scaled to increase the gap by any amount.

We can therefore simplify our task to identifying a vector a such that, for

every point x′ in class A, we have

i

i ia x aå ¢ ³ +0 1

and that, for every point x″ in class B, we have

i

i ia x aå ² £ -0 1

Let’s introduce a positive variable for each of the data points, say y i′ for

each point of class A and y i″ for each of class B. Now the inequality

∑i ai xi ≥ a0 + 1 can be enforced by requiring

¢ ³ + - å ¢

y a a x

i

i i0 1 and ¢ ³y 0

and minimizing y′ to zero. The algebra is symmetric for the points of class B.

All in all, we are led to the following optimization problem:

min i A

i

i B

iy y

Î

¢

Î

²

å å+

subject to ¢ ³ + - å ¢

y a a x

i

i i0 1 ,

¢¢ ³ - +å ²

y a x a

i

i i 0 1 ,

and

¢ ¢¢ ³y y, 0

One characteristic of this model is that if the optimal objective value

is zero, we have a hyperplane correctly separating the training set into

malignant cells and benign cells. But if the value is non-zero, it means that

the set is not separable by a hyperplane and so more complex techniques

are required.

Chapter 2 Linear Continuous ModeLs

60

2.5.2 Executable Model

Let’s translate this into the executable model seen in Listing 2-7. It accepts

two sets of data points with any number of features, classified by some

expert into classes A and B. After defining the potential deviation from

the hyperplane of sets A and B on lines 4 and 5, we define the variable

that will hold the hyperplane on line 6. Note that we need this hyperplane

later on, to do the classification of the unknown points. Note also that the

coefficients could be restricted to be in any interval containing zero. It is

simple to scale all coefficients of a plane to have its algebraic expression

reside on whatever interval we choose, as long as it includes zero.

The constraints at lines 8 and 10 set up the offset of each point to the

hyperplane which the objective function will attempt to minimize to zero.

Listing 2-7. Identification of the Classifying Hyperplane (features.py)

1 def solve\_classification(A,B):

2 n,ma,mb=len(A[0]),len(A),len(B)

3 s = newSolver('Classification')

4 ya = [s.NumVar(0,99,'') for \_ in range(ma)]

5 yb = [s.NumVar(0,99,'') for \_ in range(mb)]

6 a = [s.NumVar(-99,99,'') for \_ in range(n+1)]

7 for i in range(ma):

8 s.Add(ya[i] >= a[n]+1-s.Sum(a[j]\*A[i][j] for j in

range(n)))

9 for i in range(mb):

10 s.Add(yb[i] >= s.Sum(a[j]\*B[i][j] for j in range(n))-

a[n]+1 )

11 Agap = s.Sum(ya[i] for i in range(ma))

12 Bgap = s.Sum(yb[i] for i in range(mb))

13 s.Minimize(Agap+Bgap)

14 rc = s.Solve()

15 return rc,ObjVal(s),SolVal(a)

Chapter 2 Linear Continuous ModeLs

61

The reader might feel a little uncomfortable about this model and

here is why, at least partly: This is a model where we do not care about the

optimal value, but only whether it is zero or not. The decision variables

(and we already discussed why this expression is such a misnomer) are not

deciding anything. The set of y variables has no real interpretation other

than it represents by how much a point violates a linear inequality. And

finally, the only part of the solution we extract, the hyperplane, is not used

yet. It will only be used later on, in a different program trying to classify a

new point as belonging to class A or B. We have moved, with this model, to

a higher abstract plane than ever before.

2.5.2.1 Variations

There are at least three directions we can go from this model.

• The first is to add constraints to increase the quality

of the returned hyperplane. For example, we could

require that it not only separates the two sets, but that it

is, in some sense, as far from one set as from the other.

If the training set is well-chosen, this will ensure that

we minimize erroneous classifications later on. This

is known as maximizing the margin and we will tackle

this problem in a later chapter.

• The second direction to pursue is what to do when the

optimal value is not zero; that is, when the two sets are

not separable by a hyperplane. They may be separable

by a nonlinear curve. This question is complex and

multiple approaches have been tried, but most rely on

knowing something additional about the data. We will

not consider it.

• The final improvement would be to consider

classification into multiple classes. We will consider

this in a later chapter.

Chapter 2 Linear Continuous ModeLs

63© Serge Kruk 2018

S. Kruk, Practical Python AI Projects, https://doi.org/10.1007/978-1-4842-3423-5\_3

CHAPTER 3

Hidden Linear

Continuous Models

In this chapter we do violence to some problems to reveal their inner

structure. The focus is on problems which, at first glance, may not seem

to be of the continuous linear variety yet can be marshalled into that form

with a handful of creative alterations. The key is to ensure a one-to-one

correspondence between the original and the altered problems so that we

can retrieve a solution to the original from a solution to the alteration.

The main reason for massaging problems in this way is that continuous

linear solvers have become so fast that they can handle models with

hundreds of thousands of variables and constraints. Therefore, if a

problem can be modeled in that manner, there is little practical limit on

the instance size that can be solved. As you will see later, this is not the

case with more complex models. In fact, we can write models with a few

dozen variables that no current solver can solve in a reasonable time.

The main obstacles encountered in this chapter are non-linearities

of one kind or another, but with the advantageous restriction that the

functions be considered convex. A convex function1 is one that sits “above”

1All research mathematicians agree on the labels “convex” and its opposite,

“concave,” but textbook authors for high schools in the US, ignoring thousands of

papers, journals, and research monographs, insist on “concave up” and “concave

down.”

64

any valid linear approximation to it. In one dimension, algebraically, f is

convex at point x0 if

f x h f x f x h0 0 0+( ) ³ ( ) + ( )¢

Geometrically, it looks like Figure 3-1, with a first-order approximatio