ETC3250 - Project

Souhaib Ben Taieb, Di Cook, Rob Hyndman

Can we tell what is in a painting by the colors of the pixels?

The project for this course is to analyse the happy paintings by Bob Ross. This was the subject of the 538 post, "A Statistical Analysis of the Work of Bob Ross".

We have taken the painting images from the sales site, read the images into R, and resized them all to be 20 by 20 pixels. Each painting has been classified into one of 8 classes based on the title of the painting. This is the data that you will work with.

It is provided in wide and long form. Long form is good for making pictures of the original painting, and the wide form is what you will need to use for fitting the classification models. In wide form, each row corresponds to one painting, and the rgb color values at each pixel are in each column. With a 20×20 image, this leads to $400 \times 3 = 1200$ columns.

There are 241 paintings in the full data set. We have given you a training set of 178 paintings. Your job is to build the best classifier that you can with this training data, and use it to predict the data that we have not labelled. We have the classes for this test data, and can provide you with the error for your predictions.

Here are three of the original paintings in the collection, labelled as "scene", "water", "flowers":

Here is a quick look at the data produced from the paintings, and some code for making plots.

Wide form

Wide form dimensions:

```
paintings <- read.csv("../data/paintings-train.csv", stringsAsFactors=FALSE)
dim(paintings)
```

```
## [1]  178 1203
```

Number of paintings in each class:

```
table(paintings$class)
```

```
##
##         cold         dusk      flowers impressions         oval        scene
##           23           30           22           17            5           28
##        trees        water
##           18           35
```

First few columns and rows:

```
options(digits=2)
paintings[1:5, 1:11]
```

```
##   id                name   class    r1   g1   b1    r2   g2   b2   r3   g3
## 1  1        crimson-oval    oval 0.616 0.56 0.48 0.725 0.66 0.60 0.74 0.68
## 2  2          sunflowers flowers 0.922 0.99 1.00 1.000 0.98 0.94 1.00 0.96
## 3  4        lonely-cabin    cold 0.086 0.15 0.15 0.170 0.21 0.22 0.39 0.44
## 4  5   crimson-mountains   scene 0.325 0.67 0.81 0.353 0.60 0.66 0.60 0.66
## 5  6 reflections-of-fall   water 0.000 0.34 0.62 0.024 0.38 0.73 0.00 0.42
```

Long form

Long form dimensions:

```
paintings_long <- read.csv("../data/paintings-long-train.csv", stringsAsFactors=FALSE)
dim(paintings_long)
```

```
## [1] 71200     9
```

First few rows:

```
head(paintings_long)
```

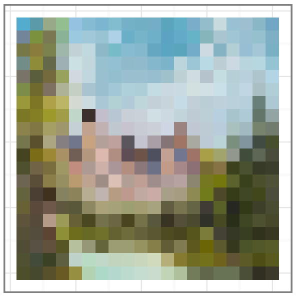
```
##   x y    r    g    b       h         name id class
## 1 1 1 0.62 0.56 0.48 #9D8E7B crimson-oval  1  oval
## 2 2 1 0.73 0.66 0.60 #B9A999 crimson-oval  1  oval
## 3 3 1 0.74 0.68 0.62 #BEAE9F crimson-oval  1  oval
## 4 4 1 0.72 0.65 0.60 #B7A798 crimson-oval  1  oval
## 5 5 1 0.75 0.69 0.63 #C0B0A1 crimson-oval  1  oval
## 6 6 1 0.76 0.70 0.64 #C2B2A3 crimson-oval  1  oval
```

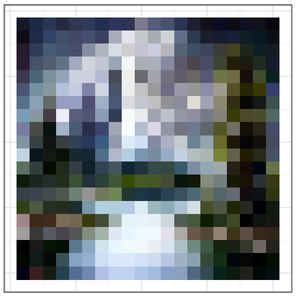
Plots of the long form data, for the same three paintings:

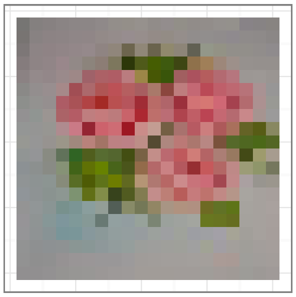
```
library(ggplot2)
library(dplyr)
df <- filter(paintings_long, id == 5)
qplot(x, -y, data=df, fill=h, geom="tile") + scale_fill_identity(labels=df$h) + theme_bw() +
    theme(axis.title.x = element_blank(),
          axis.title.y = element_blank(),
          axis.text.x = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks = element_blank())
```

```
df <- filter(paintings_long, id == 140)
qplot(x, -y, data=df, fill=h, geom="tile") + scale_fill_identity(labels=df$h) + theme_bw() +
    theme(axis.title.x = element_blank(),
          axis.title.y = element_blank(),
          axis.text.x = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks = element_blank())
df <- filter(paintings_long, id == 167)
qplot(x, -y, data=df, fill=h, geom="tile") + scale_fill_identity(labels=df$h) + theme_bw() +
    theme(axis.title.x = element_blank(),
          axis.title.y = element_blank(),
          axis.text.x = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks = element_blank())
paintings$class[paintings$id %in% c(5,140,167)]
```

```
## [1] "scene"   "water"   "flowers"
```







Tasks

1. Provide some exploratory descriptive summaries of the data, that illustrate how the types of paintings differ, or are similar, and point out unusual paintings that don't fit their group.
2. Create the most accurate classifier that you can for the data, as measured by the test data. This will require submitting your predictions to a specially created kaggle site.
3. Write a 3–5 page report summarizing your approach to building the classifier, and your findings about the data.

Comments

- The labels for the paintings have been automatically created from the painting names. It is possible that a painting might be better labelled as a different class.
- It might be good to create new variables to help you build a better classifier; for example, the average of pixels in corners.
- You will need to submit at least one entry to the kaggle site created for testing the classifier.

Deadlines:

- Oct 14: Initial overview of data, summary statistics, plots that illustrate differences between types of paintings and unusual paintings.
- Oct 21: At least one submission of predictions to kaggle. Instructions on how to do this will come later.
- Nov 11: Final report due.

Grading:

- Total points: 40
- Accuracy of classifier: 10
- Report: 22
- Met deadlines: 8