



MONASH
University

MONASH
BUSINESS
SCHOOL

Human vs. Computer

Shuofan Zhang

5/28/2018

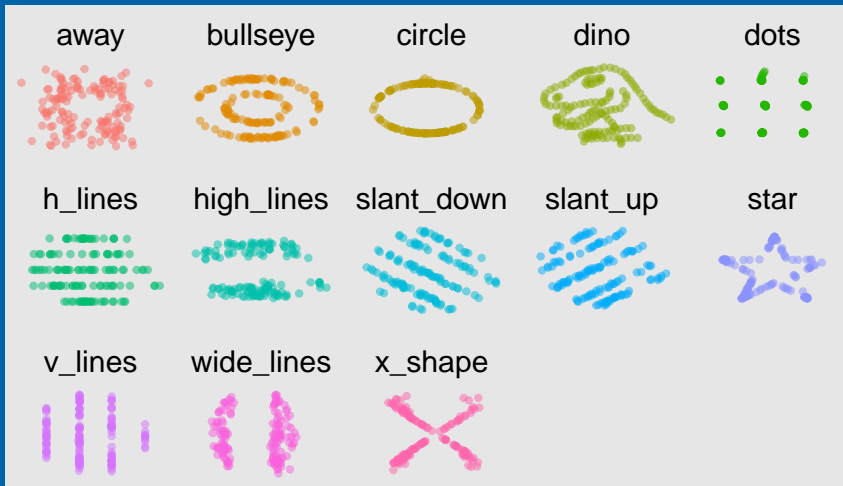
Reminder of the first presentation

Teach the computer to read residual plots

A major component used to diagnose model fits is a plot of the residuals. Residual plots are used to assess:

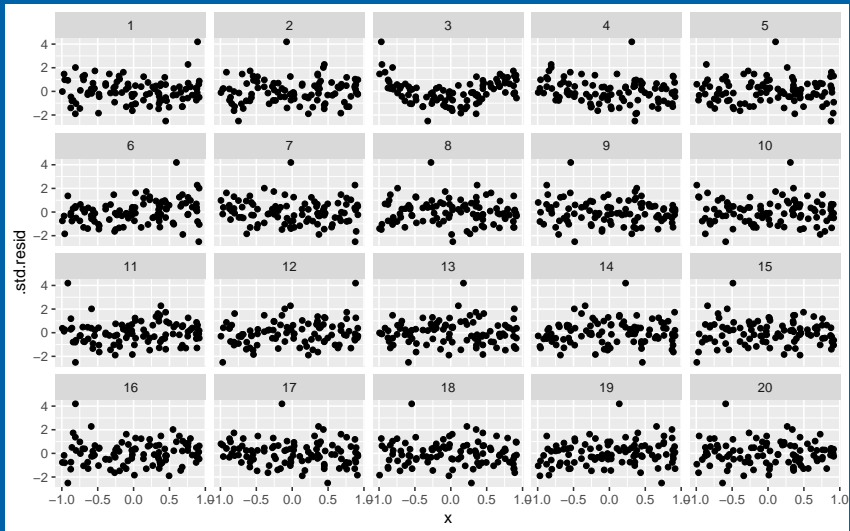
- Gauss-Markov assumption
- Heteroskedasticity
- Clumps of outliers
- ...

Why plots?



$$E(x) = 54.3, E(y) = 47.8, sd(x) = 16.8, sd(y) = 26.9, r =$$

Visual inference



Visual inference

- Plot of data is a test statistic
- Type of plot determines null hypothesis, e.g. residuals vs fitted scatterplot would imply H_o : no relationship, vs H_a : some relationship
- Human visual system evaluates lineup of data plot in field of null plots
- If data plot is “identified” as different from null, H_o is rejected
- Combining results from multiple observers enables p -value calculation

Aside: Computers can't tell difference between blueberry muffins and chihuahuas

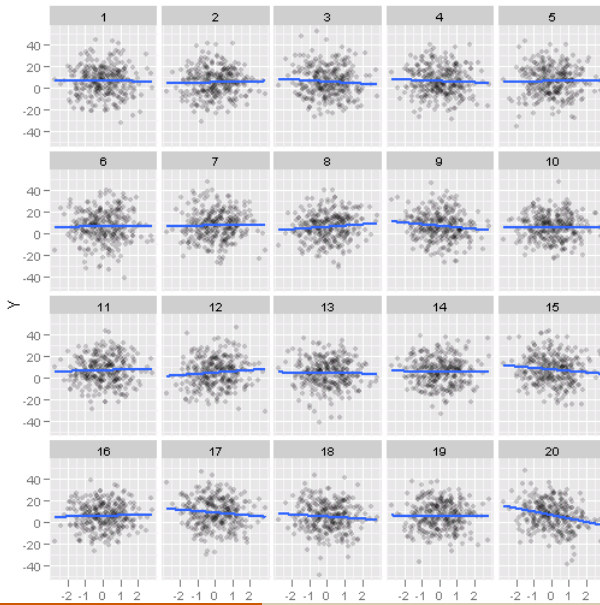


Figure 1: Computers can't tell difference between blueberry

Amazon Turk study

- Majumder et al (2013) conducted a large study to compare the performance of the lineup protocol, assessed by human evaluators, in comparison to the classical test
- Experiment 2 examined $H_o : \beta_k = 0$ vs $H_a : \beta_k \neq 0$ assessing the importance of including variable k in the linear model, conducted with a t -test, and also lineup protocol
- 70 lineups of size 20 plots
- 351 evaluations by human subjects
-
- Trained deep learning model will be used to classify plots from this study. Accuracy will be compared with results by human subjects.

Example lineup from experiment 2

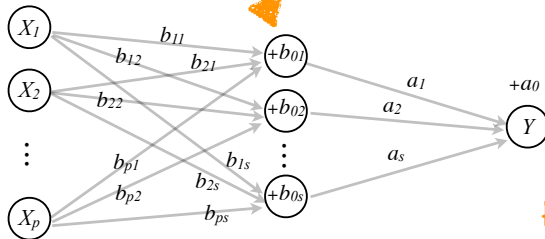


Convolutional neural network (convnets)

- Computer vision has advanced substantially
- If we can train a computer to read residual plots we can have it process a lot more data, than a human can manage.

How convnets works: Diagram of regression

➤ Add a **hidden layer** with s units



Fitted model

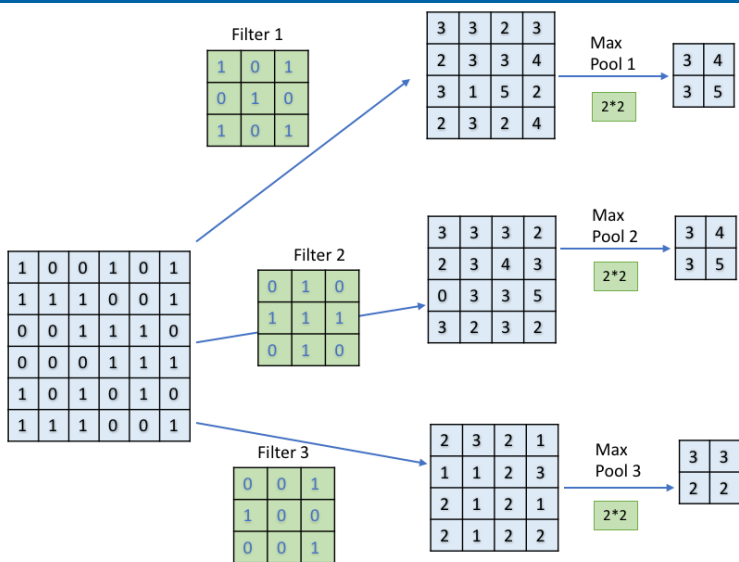
$$\hat{Y} = a_0 + \sum_{j=1}^s (a_j (b_{0j} + \sum_{i=1}^p b_{ij} X_i))$$

Figure 3: How regression works

How convnets works: R code

```
library(keras)
model <- keras_model_sequential() %>%
  layer_conv_2d(filters = 32, kernel_size = c(3, 3),
    activation = "relu",
    input_shape = c(150, 150, 1)) %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3, 3),
    activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 128, kernel_size = c(3, 3),
    activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_flatten() %>%
  layer_dense(units = 512, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

How convnets works: Diagram of convolution and max pooling



First Experiment

H_0 : There are no relationships between the two variables.

H_1 : There is linear relationship between the two variables where all Gauss-Markov assumptions are met.

Steps: 1. Simulate data (X, Y) from the null and the alternative models 2. Generate scatter plots of X and Y 3. Save scatter plots as 150×150 pixels images 4. Train a deep learning classifier to recognise the patterns from two groups 5. Test the model's performance on new data and compute the accuracy

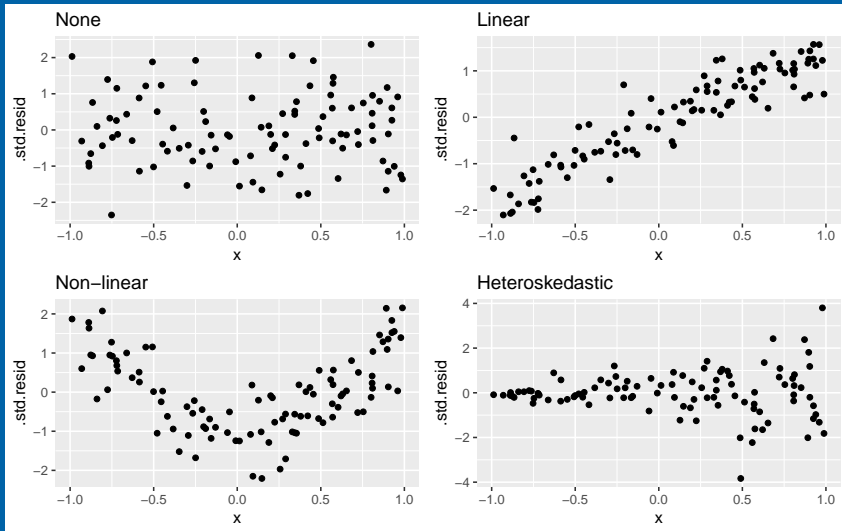
First Experiment: Data simulation

The model:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, \dots, n$$

- $X \sim N[0, 1]$
- $\beta_0 = 0$
- $\beta_1 \sim U[-10, -0.1] \cup [0.1, 10]$
- $\varepsilon \sim N(0, \sigma^2)$ where $\sigma \sim U[1, 12]$
- $n = U[50, 500]$

First Experiment: Scatter plots



Comparison with human subject experiments

Timeline

Date	Component
Apr 27	Deep learning model trained
May 4	Classification of new residual plots with model and results summarised
May 18	Comparison with Turk studies
May 24	Refinements made, final summaries written
May 31	Thesis finalised

Materials

- The thesis, code and data is available on the github repository <https://github.com/shuofan18/ETF5550>
- Software used to conduct this research is R, Tensorflow, keras, tidyverse