MONASH University

MONASH BUSINESS SCHOOL

Human vs. Computer

Shuofan Zhang

5/28/2018

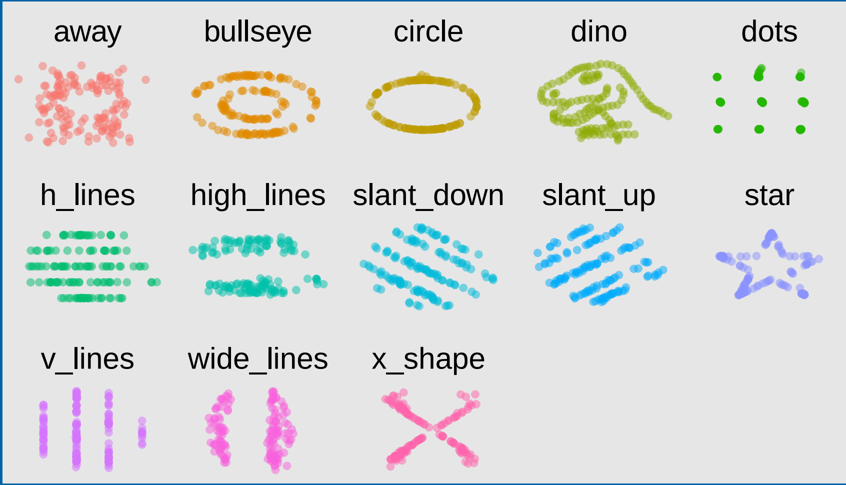
Reminder of the first presentation

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:

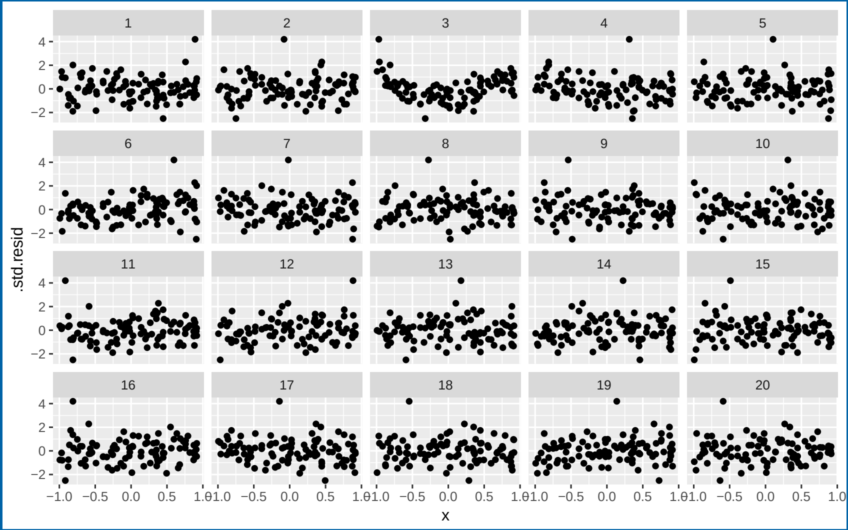
- Gauss-Markov assumption
- Heteroskedasticity
- Clumps of outliers
- ...

Why plots?



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

Visual inference



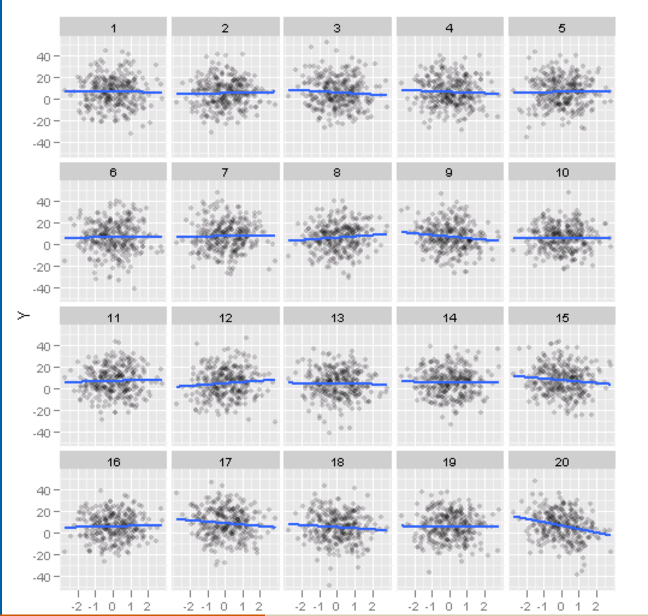
Aside: Computers can't tell difference between blueberry muffins and chihuahuas

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 comaprison 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: 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: Model structure

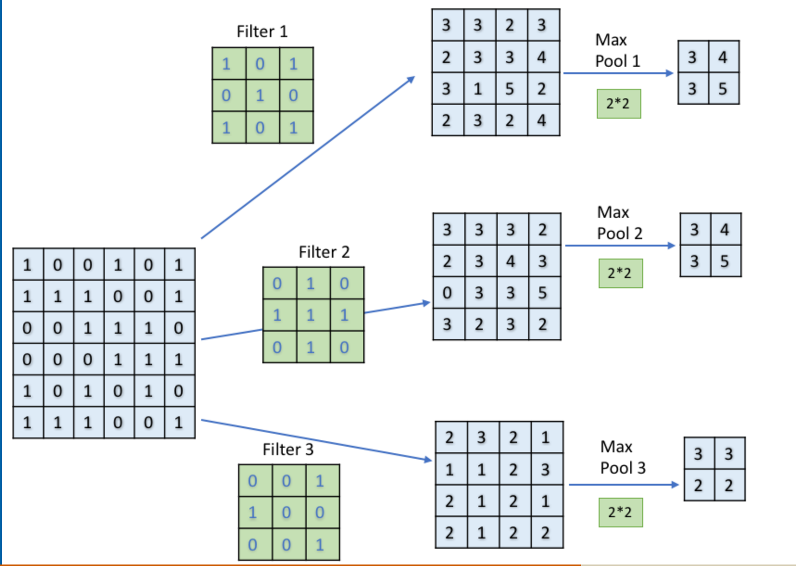
Model

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_3 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_1 (Flatten)	(None, 36992)	0
dense_1 (Dense)	(None, 512)	18940416
dense_2 (Dense)	(None, 1)	513

Total params: 19,033,601
Trainable params: 19,033,601
Non-trainable params: 0

Figure 3: convnets model structure

How convnets works: Diagram of convolution and max pooling



First Experiment: Linear vs. Null

H_0: There are no relationships between the two variables. (Null) H_1: There is linear relationship between the two variables where all Gauss-Markov assumptions are met. (Linear)

Computer model procedures:

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: Computer model procedures

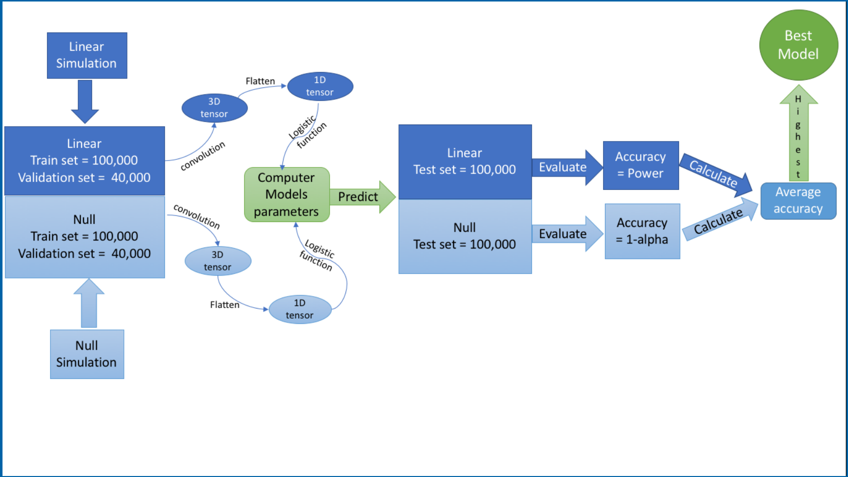


Figure 5: Procedure of computer model experiment

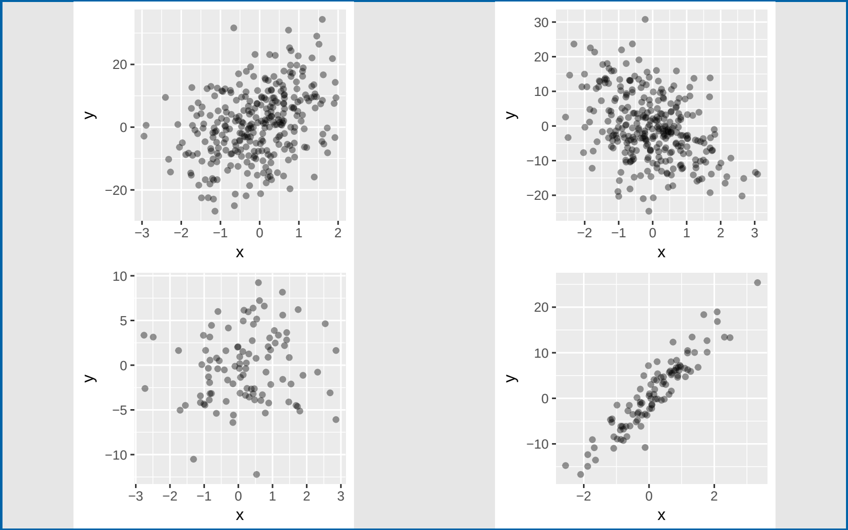
First Experiment: Data simulation

The model:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \;\; i = 1, \ldots, n$$

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

First Experiment: Examples of scatter plots



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