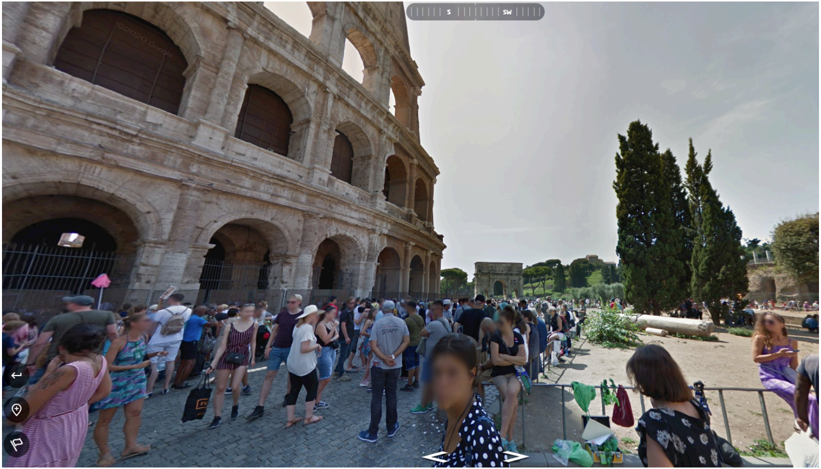
How to do OK at GeoGuessr using simple statistics

Matthew Pawley

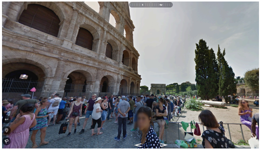
Example (a) – what country is this?



Example (b) – what country is this?

Example (c) – what country is this?

How do humans perform photo geolocation?



(a) Italy (Rome)

(b) Italy (Pisa)

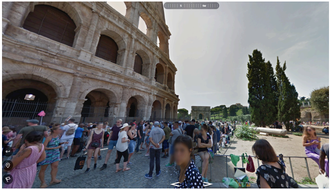
(c) South Africa

	Method	**Description**
(a)	Memorisation	"I've seen it before"
(b)	Scene matching	"It is similar to something I've seen before"
(c)	Semantic reasoning	"I used fluid logic and contextual information"

Problem formulation

$\mathcal{X}$	space of all Google Street View images
$\mathcal{Y}$	space of all the corresponding locations
$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$	training data (labelled examples)

Question: Given an image $\boldsymbol{x} \in \mathcal{X}$, what is its location $y \in \mathcal{Y}$?



$\xrightarrow{f}$ $\boxed{(41.89^\circ\text{N},\ 12.49^\circ\text{E})}$

The k-nearest-neighbours (k-NN) algorithm

Figure: A new image, $\boldsymbol{x}$, whose location y is unknown.

(a) London (b) London (c) St Petersburg

Figure: The $k = 3$ most similar training images and their (known) locations.

$\hat{y} = \text{London}$[1]

---

[1] This step is non-trivial. Hayes and Efros (2008) use 'mean shift mode'.

Hayes and Efros (2008) – k-NN performs well

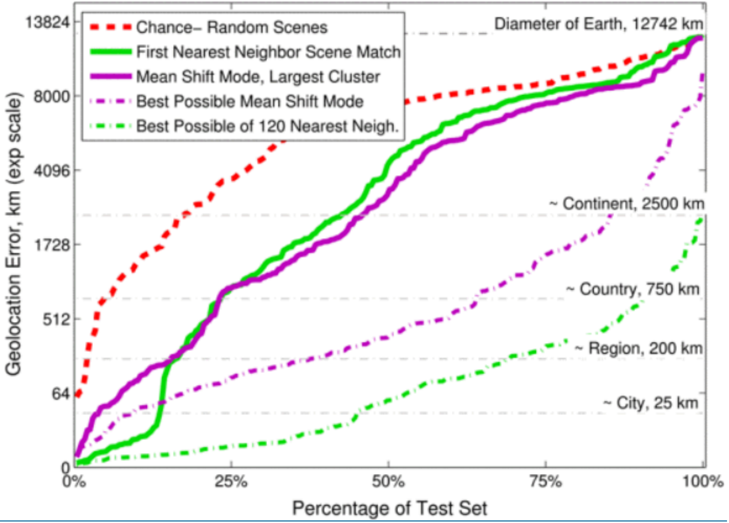


Figure: 120-NN (dashed green) does a pretty good job, much better than chance (dashed red).

Hayes and Efros (2008) – feature engineering

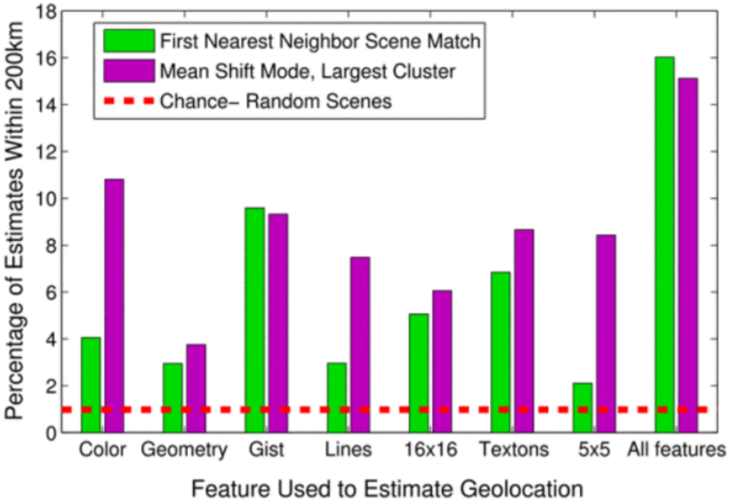


Figure: Some visual characteristics are more useful for geolocation than others.

Hypothesis spaces and types of error

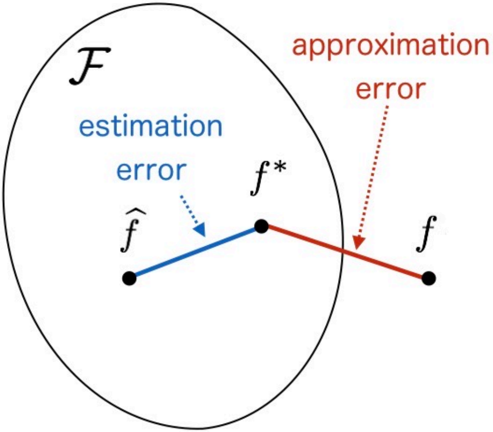
Consider k-NN with $k = 1$:

- The estimate $\hat{y}$ is just the location of the most similar image.
- The predictive model $\hat{f}$ is a piecewise constant function, i.e.

$$\hat{f} \in \mathcal{F} := \{\text{piecewise constant functions } \mathcal{X} \to \mathcal{Y}\}.$$

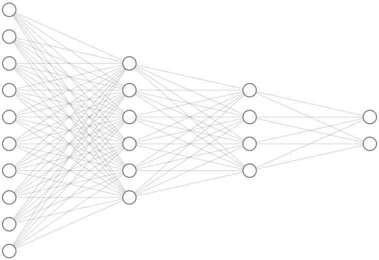
Choosing an algorithm (1-NN) $\equiv$ choosing a function space ($\mathcal{F}$)

Error decomposition



VC inequality: bounds these errors in terms of n and $\mathcal{F}$.

Neural networks



$$\mathcal{F} = \{g \circ f_l \circ \ldots \circ f_1 : f_i(\boldsymbol{x}) = \sigma(\boldsymbol{w}_i \boldsymbol{x} + \boldsymbol{b}_i)\}$$

- $\mathcal{F}$ is very big and can approximate a wide class of functions $\Rightarrow$ small approximation error
- Large $\mathcal{F}$ and small $n \Rightarrow$ likely to overfit.

What is a 'simple model'?

Scenario 1:

1. A company gives you some data.
2. You fit a linear model to it.
3. You give the company the values of the intercept and gradient.
4. The company can make predictions and you get paid.

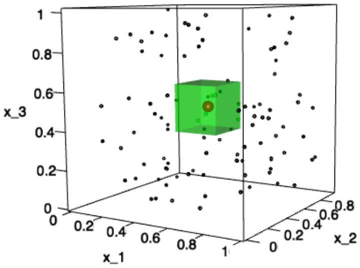
Scenario 2:

1. A company gives you some data.
2. You fit a k-NN model to it.
3. You give the company their data.
4. They are angry because you did nothing.

k-NN and the curse of dimensionality

f bdd and α-Hölder cts. $\Rightarrow$ convergence rate $= \mathcal{O}(n^{-2\alpha/(d+2\alpha)})$.

- Let $\mathcal{X} = [0, 1]^d$.
- Suppose our n training samples are distributed uniformly in $\mathcal{X}$.
- On average, the volume of the smallest hypercube containing the k-nearest-neighbours of a point $\boldsymbol{x}$ will be $V \approx k/n$.
- The side length of the hypercube is $l \approx (k/n)^{1/d}$.



k-NN and the curse of dimensionality

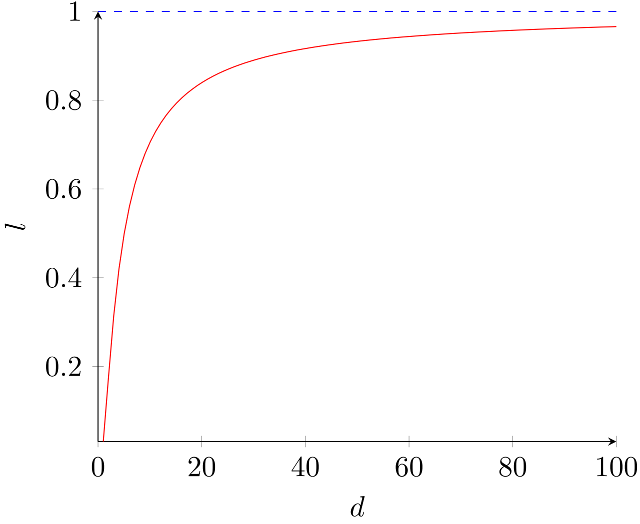


Figure: As the dimension of $\mathcal{X}$ grows, our 'local neighbourhood' becomes pretty much the whole space! (Here $k = 3$ and $n = 100$.)

Conclusions

1. Geolocating a photograph is challenging, but a simple model can do quite well.
2. Statistical learning is an illuminating and *rigorous* way of doing things.
3. If you think about it, GeoGuessr is just functional analysis.