K-Means and Support Vector Machines

Introduction and examples

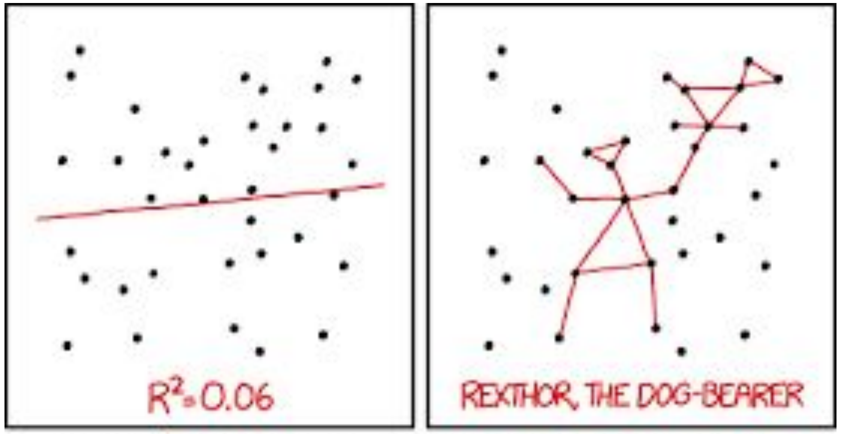
13th of November 2018

Dr Joe Fennell | Jodrell Bank Centre for Astrophysics, University of Manchester, UK | joseph.fennell@manchester.ac.uk

Part I

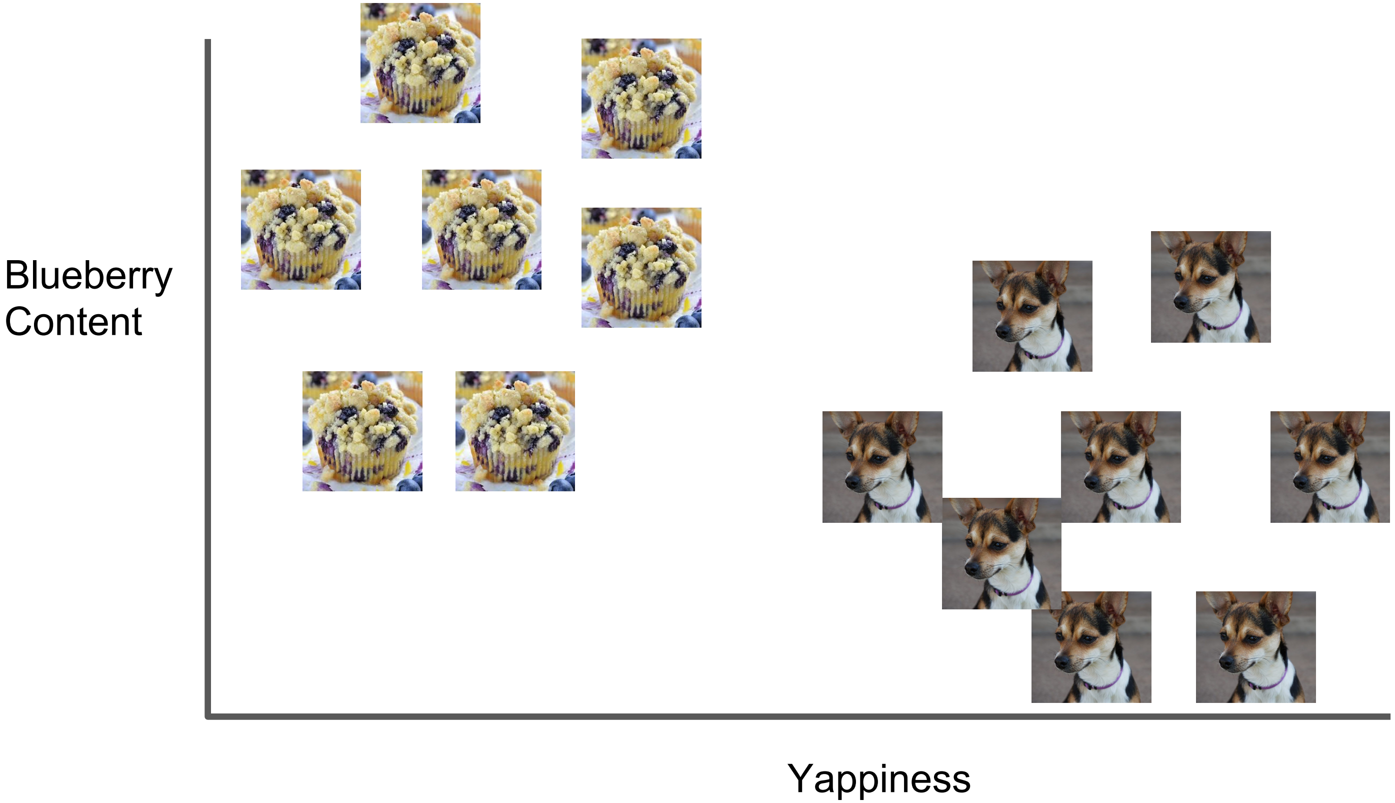
What kind of problem do I have?

OR

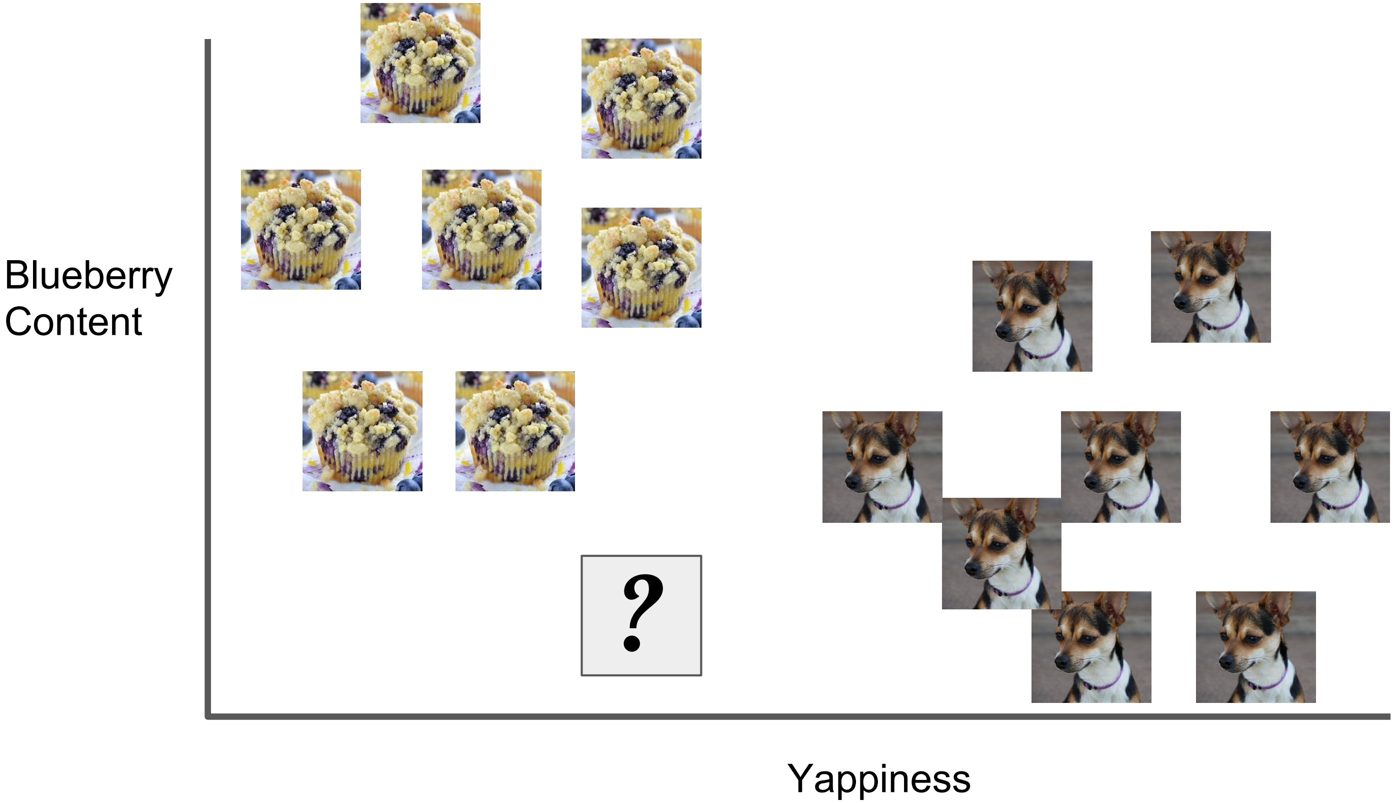


I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

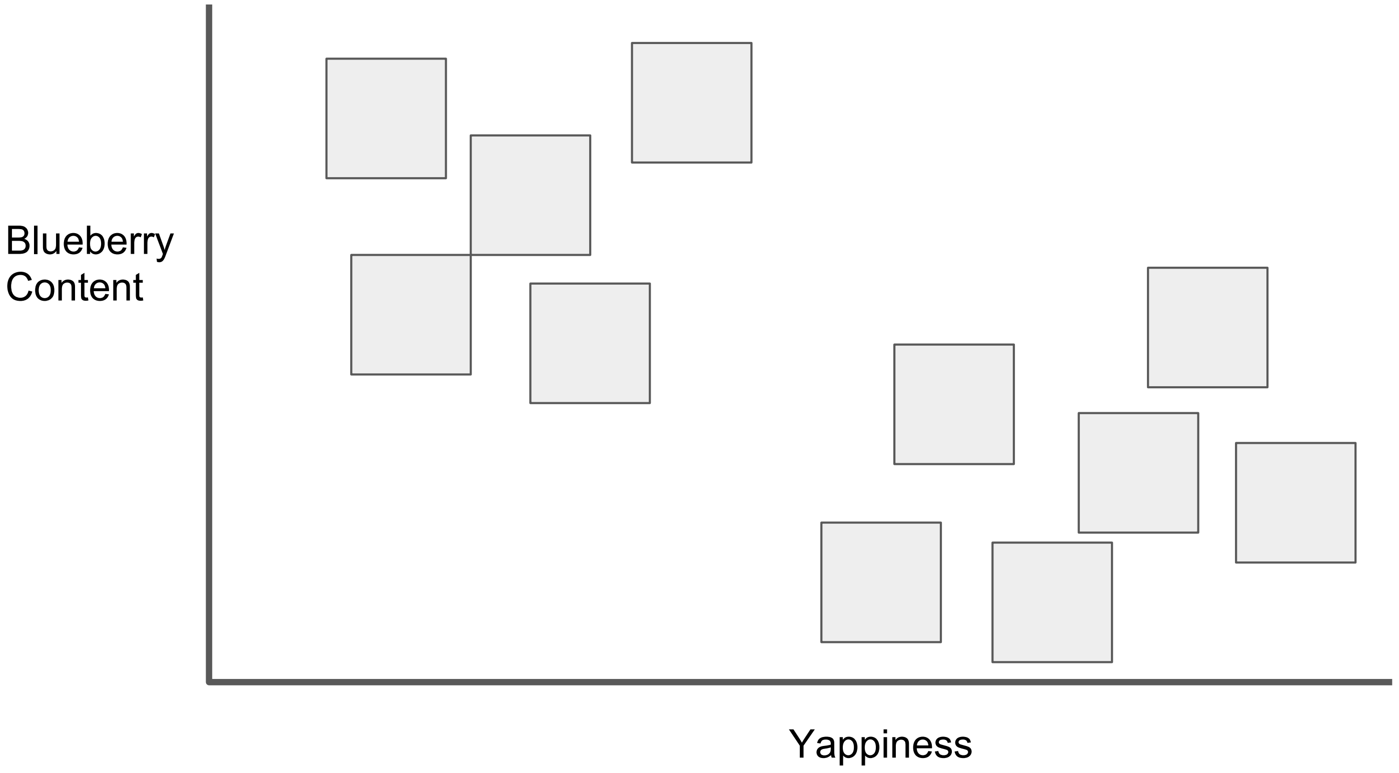
Classification Problems



Classification Problems

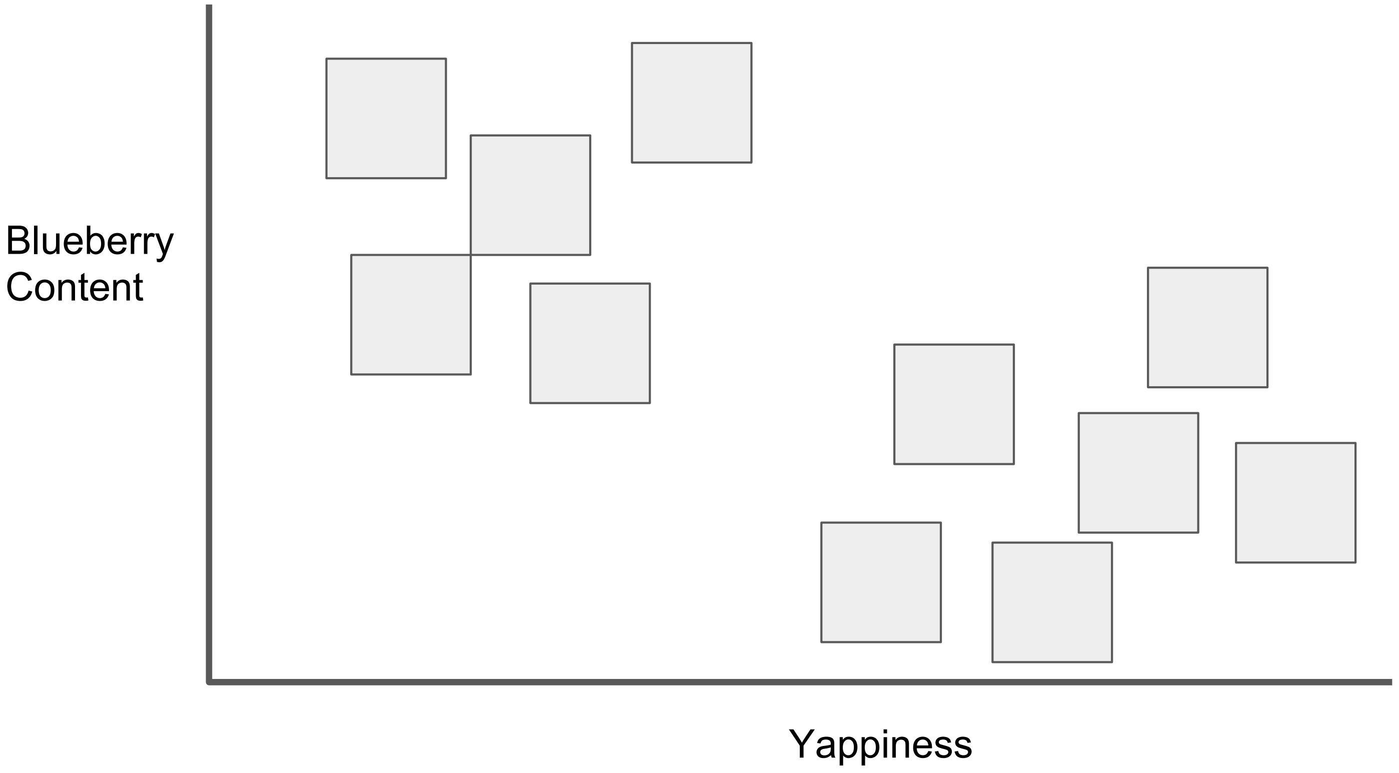


What if we have no training data?



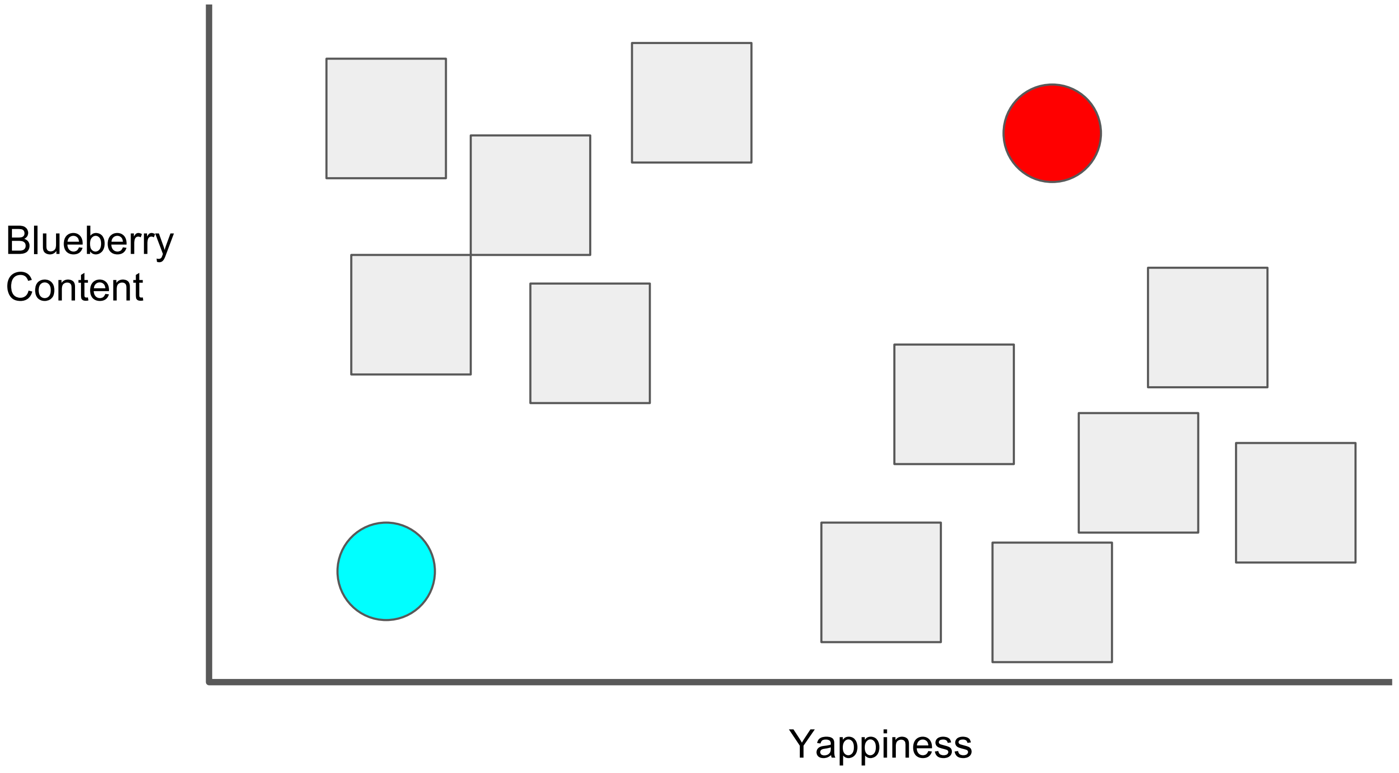
What if we have no model?

Intuitions: We know that there are two types of thing represented in the data



What if we have no model?

Intuitions: We know that there are two types of thing represented in the data

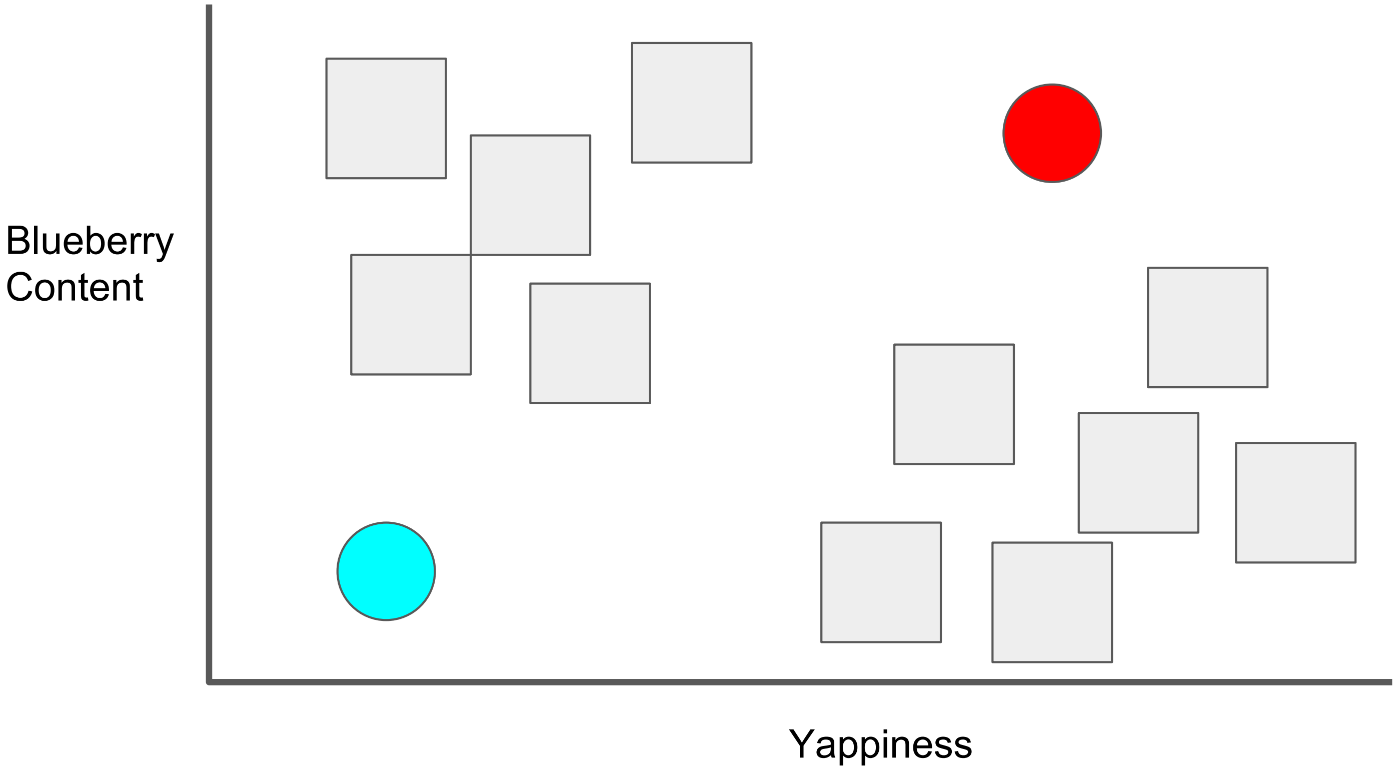


Algorithm:

1. Pick some random centre points in our parameter space

What if we have no model?

Intuitions: We know that there are two types of thing represented in the data

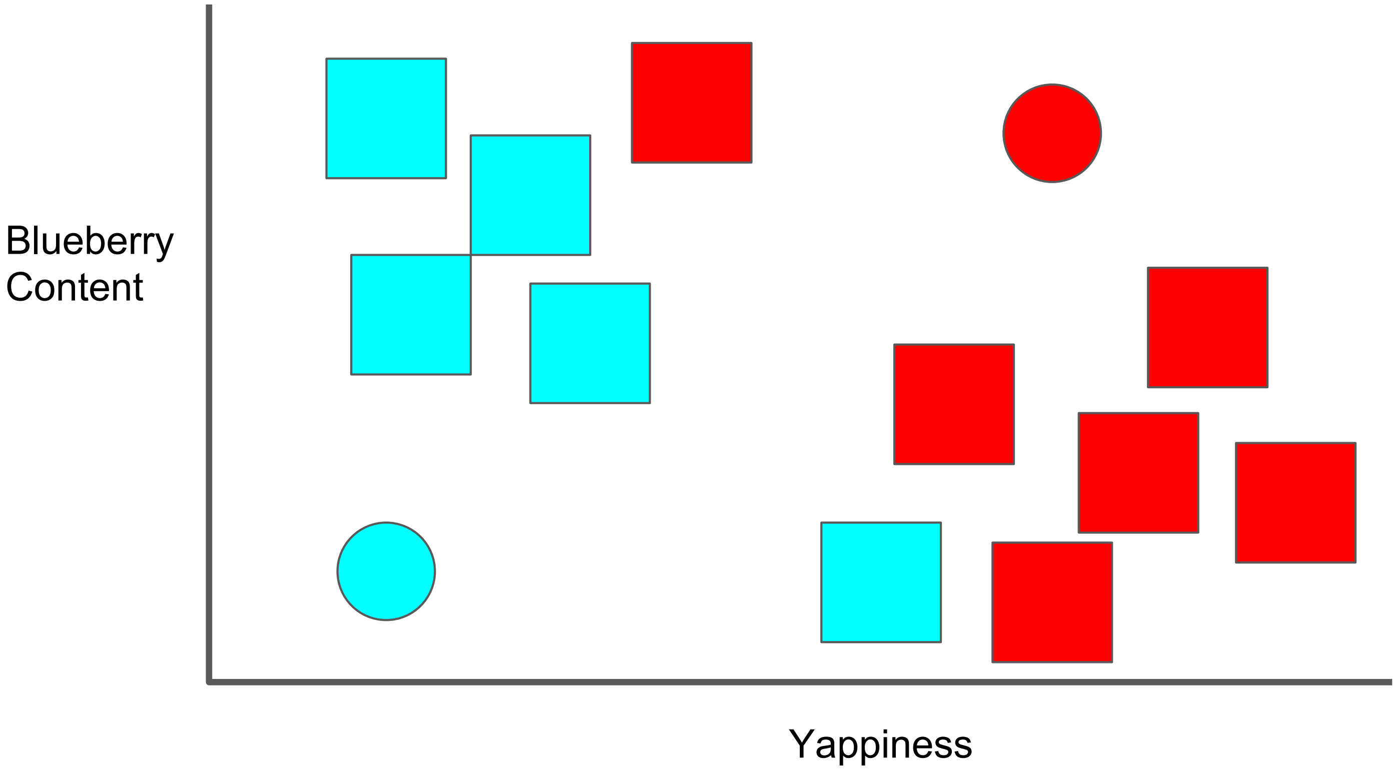


Algorithm:

1. Pick some random centre points in our parameter space
2. Calculate distance between all data and all centre points

What if we have no model?

Intuitions: We know that there are two types of thing represented in the data

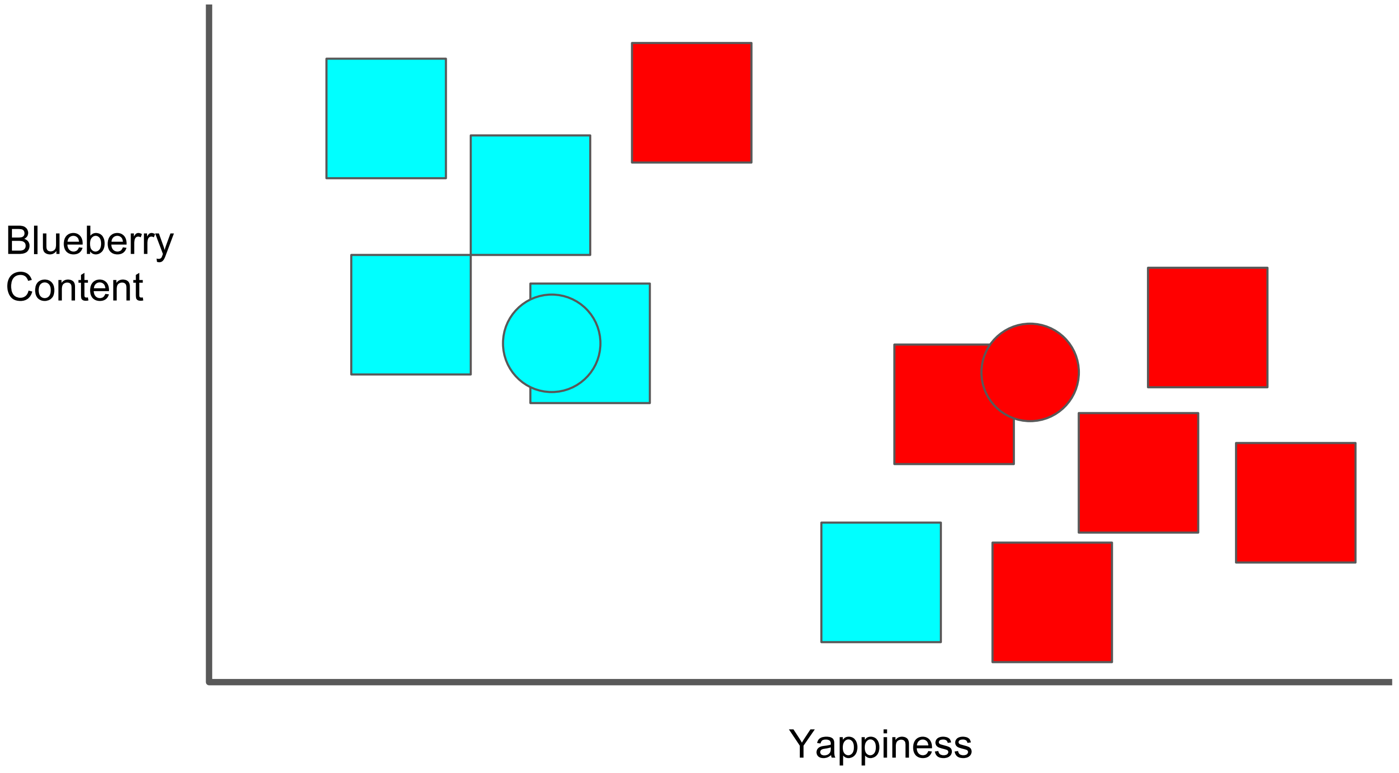


Algorithm:

1. Pick some random centre points in our parameter space
2. Calculate distance between all data and all centre points
3. Assign every datum to a centre point

What if we have no model?

Intuitions: We know that there are two types of thing represented in the data

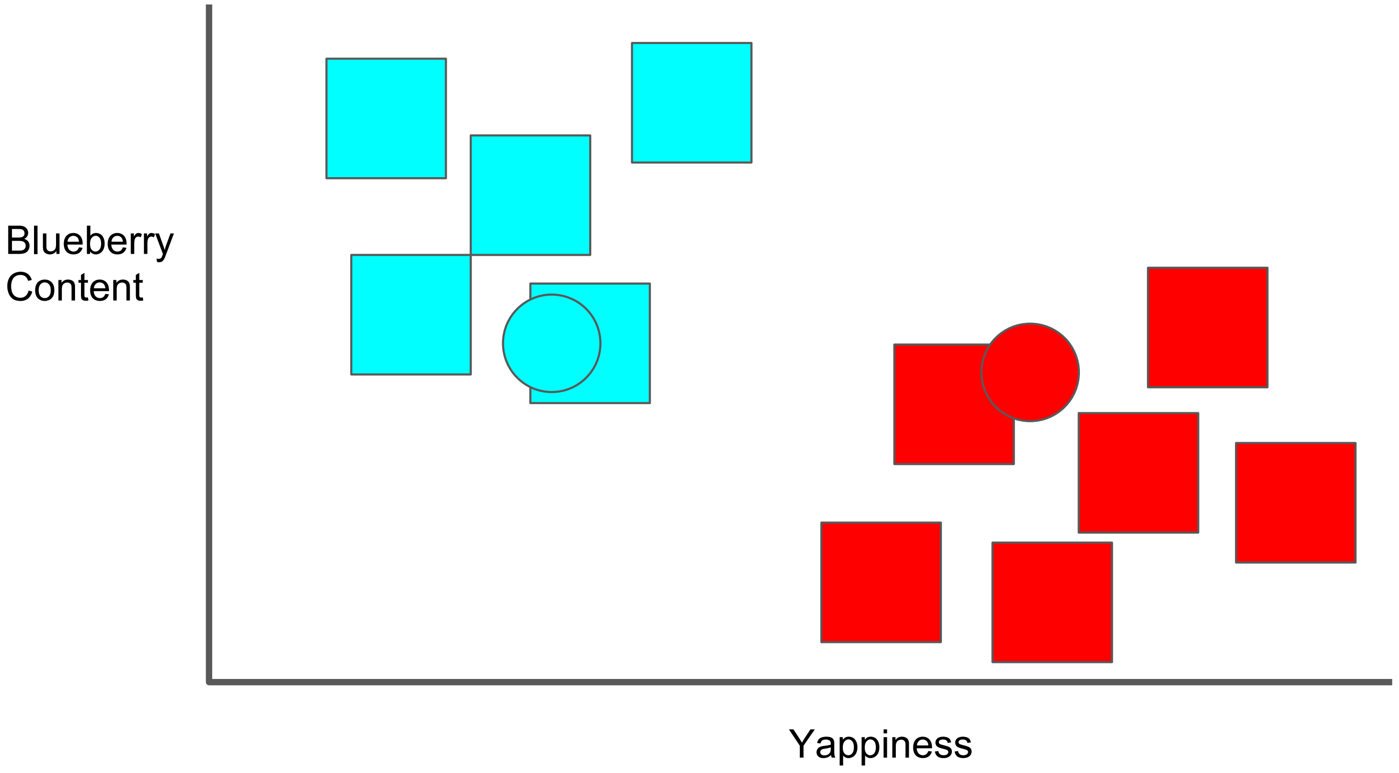


Algorithm:

1. Pick some random centre points in our parameter space
2. Calculate distance between all data and all centre points
3. Assign every datum to a centre point
4. Set centre points to means of each cluster

What if we have no model?

Intuitions: We know that there are two types of thing represented in the data

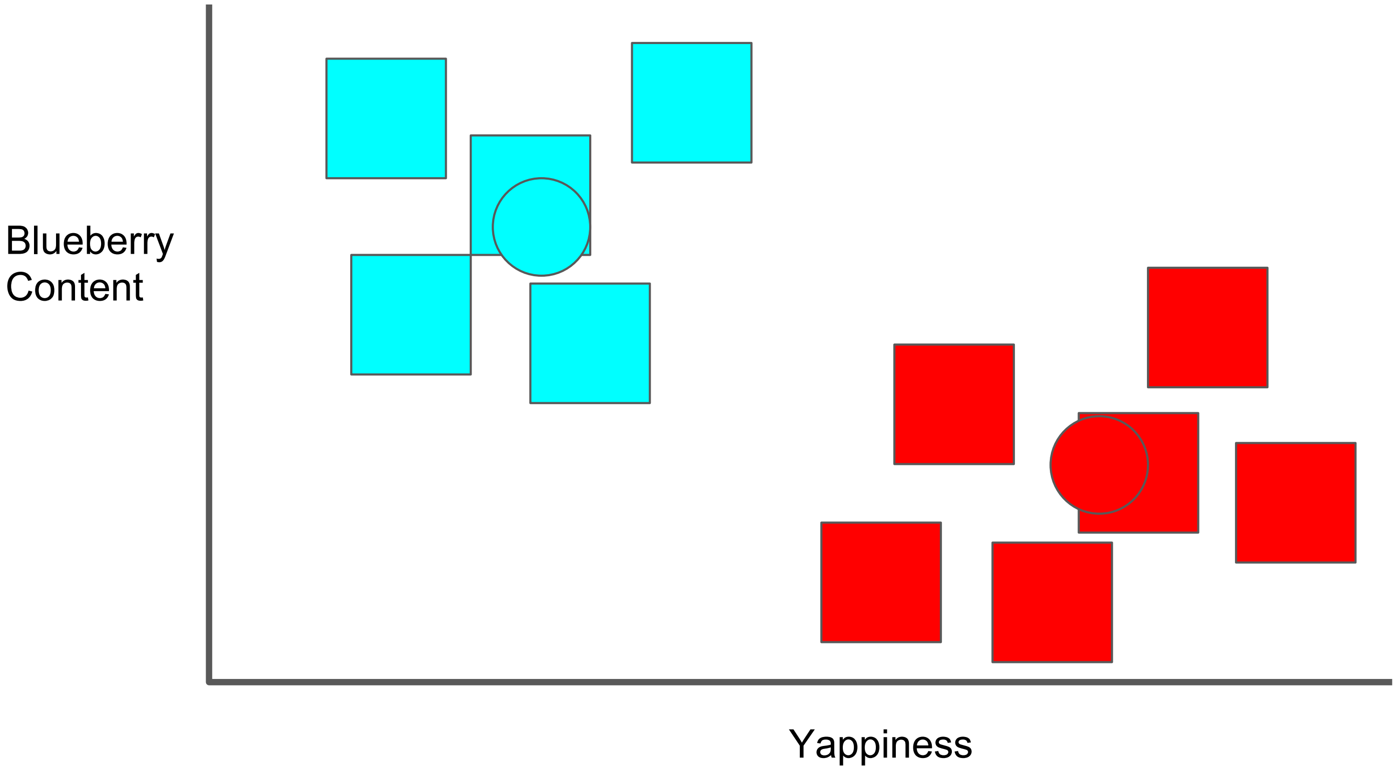


Algorithm:

1. Pick some random centre points in our parameter space
2. Calculate distance between all data and all centre points
3. Assign every datum to a centre point
4. Set centre points to means of each cluster
5. Repeat

What if we have no model?

Intuitions: We know that there are two types of thing represented in the data



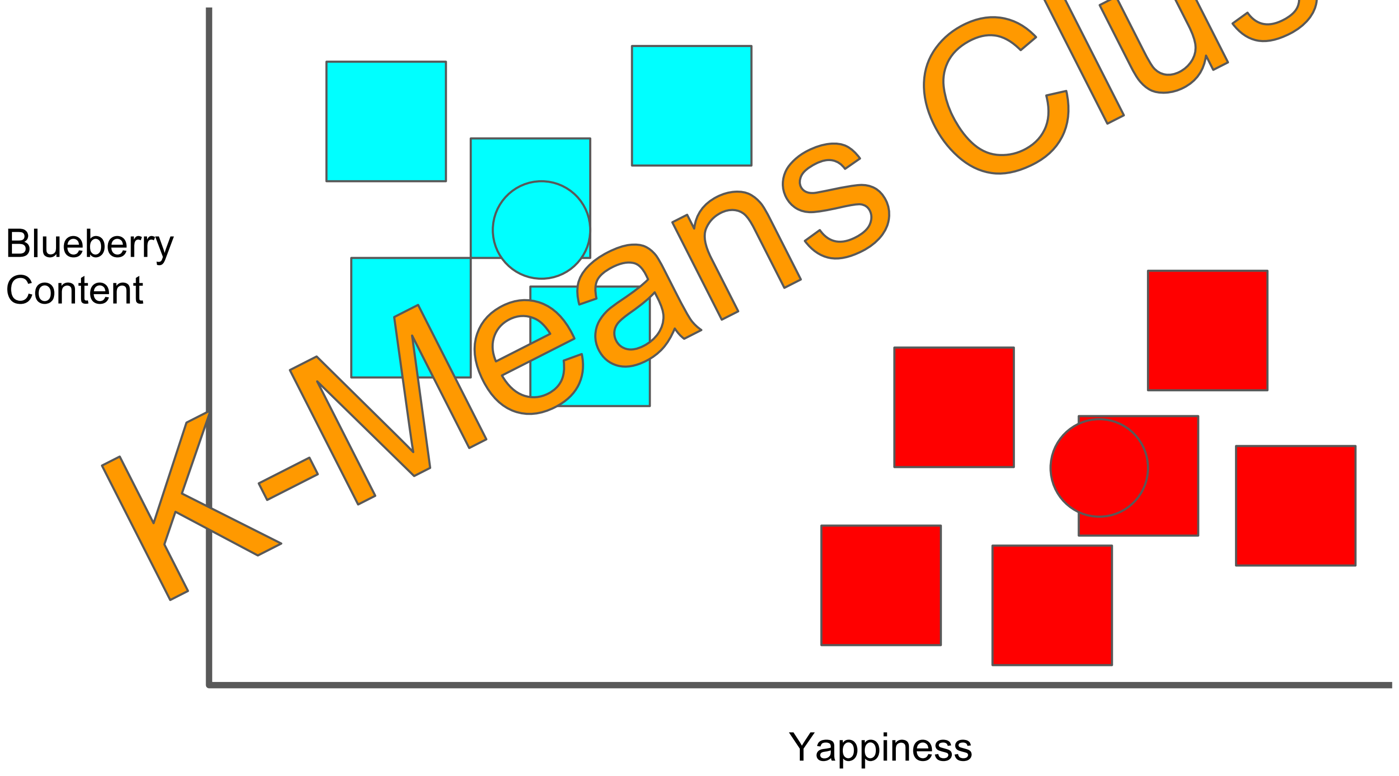
Algorithm:

1. Pick some random centre points in our parameter space
2. Calculate distance between all data and all centre points
3. Assign every datum to a centre point
4. Set centre points to means of each cluster
5. Repeat

What if we have no model?

K-Means Clustering

Intuitions: We know that there are two types of thing represented in the data



Algorithm:

1. Pick some random centre points in our parameter space
2. Calculate distance between all data and all centre points
3. Assign every datum to a centre point
4. Set centre points to means of each cluster
5. Repeat

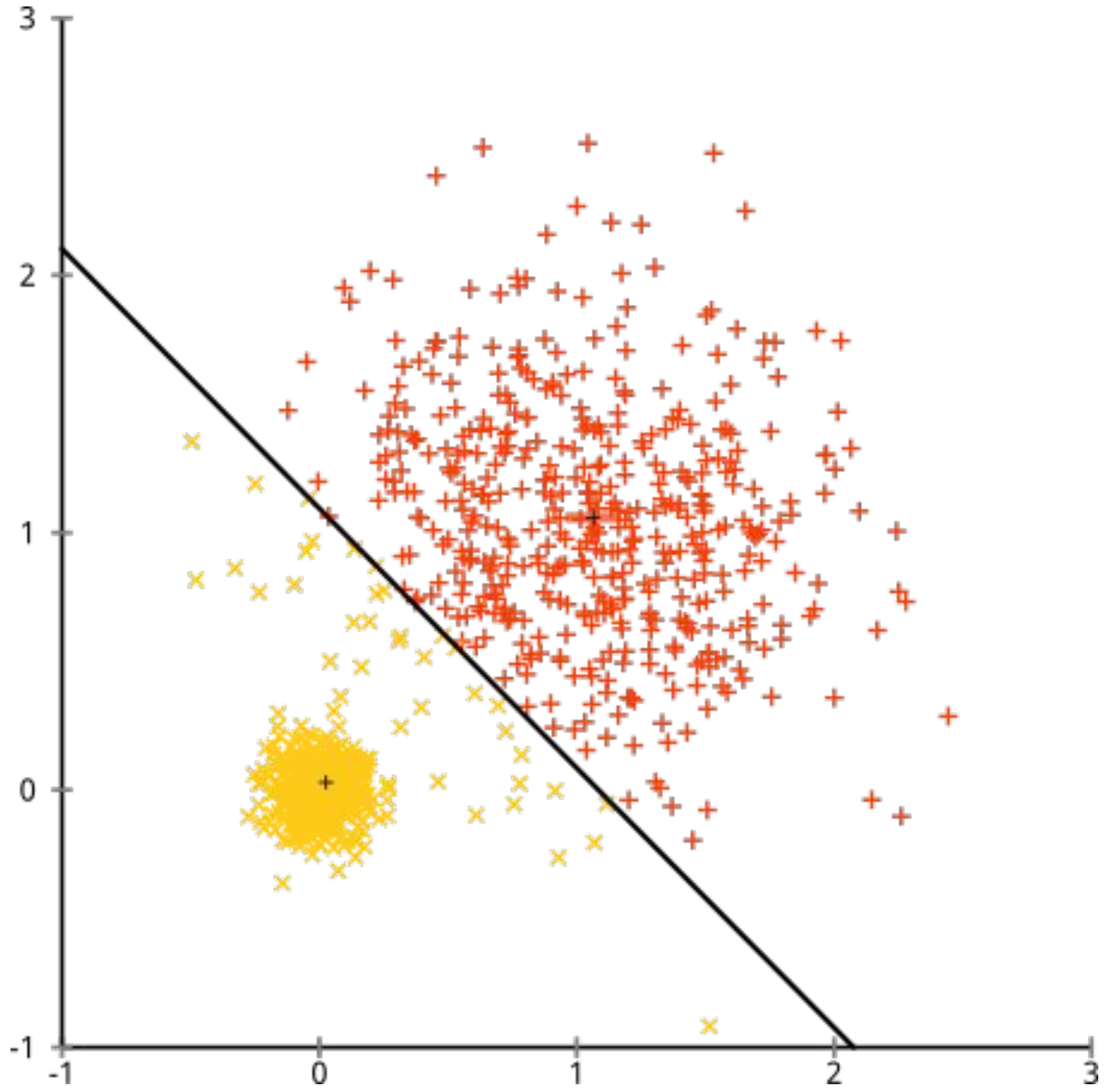
MANCHESTER 1824
The University of Manchester

K-Means issue #1

Sensitive to scale

Performs badly where parameters have differing variance

https://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means?answertab=oldest#tab-top



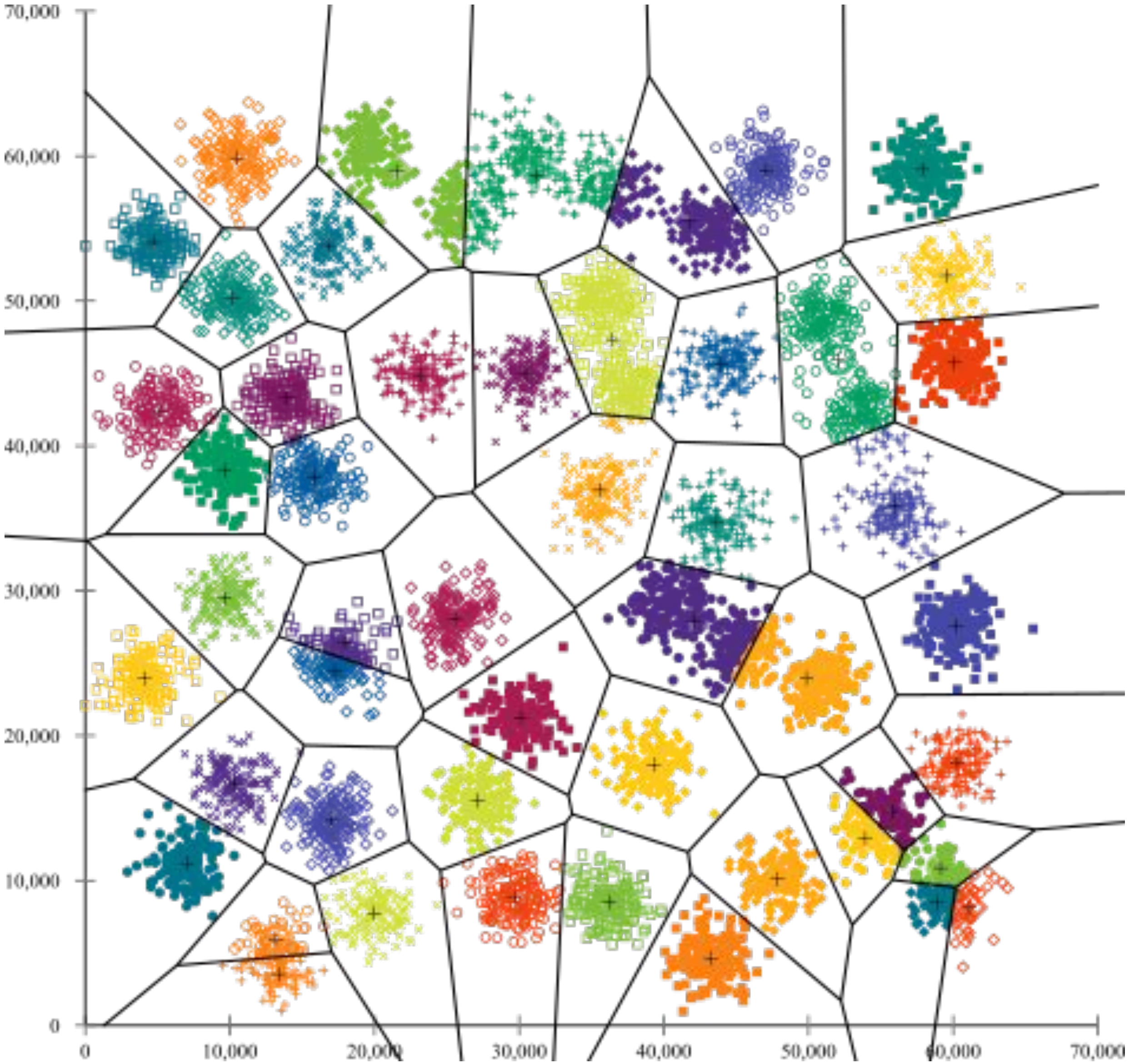
Dr Joe Fennell | Jodrell Bank Centre for Astrophysics, University of Manchester, UK | joseph.fennell@manchester.ac.uk

https://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means?answertab=oldest#tab-top

K-Means issue #2

Local Minima

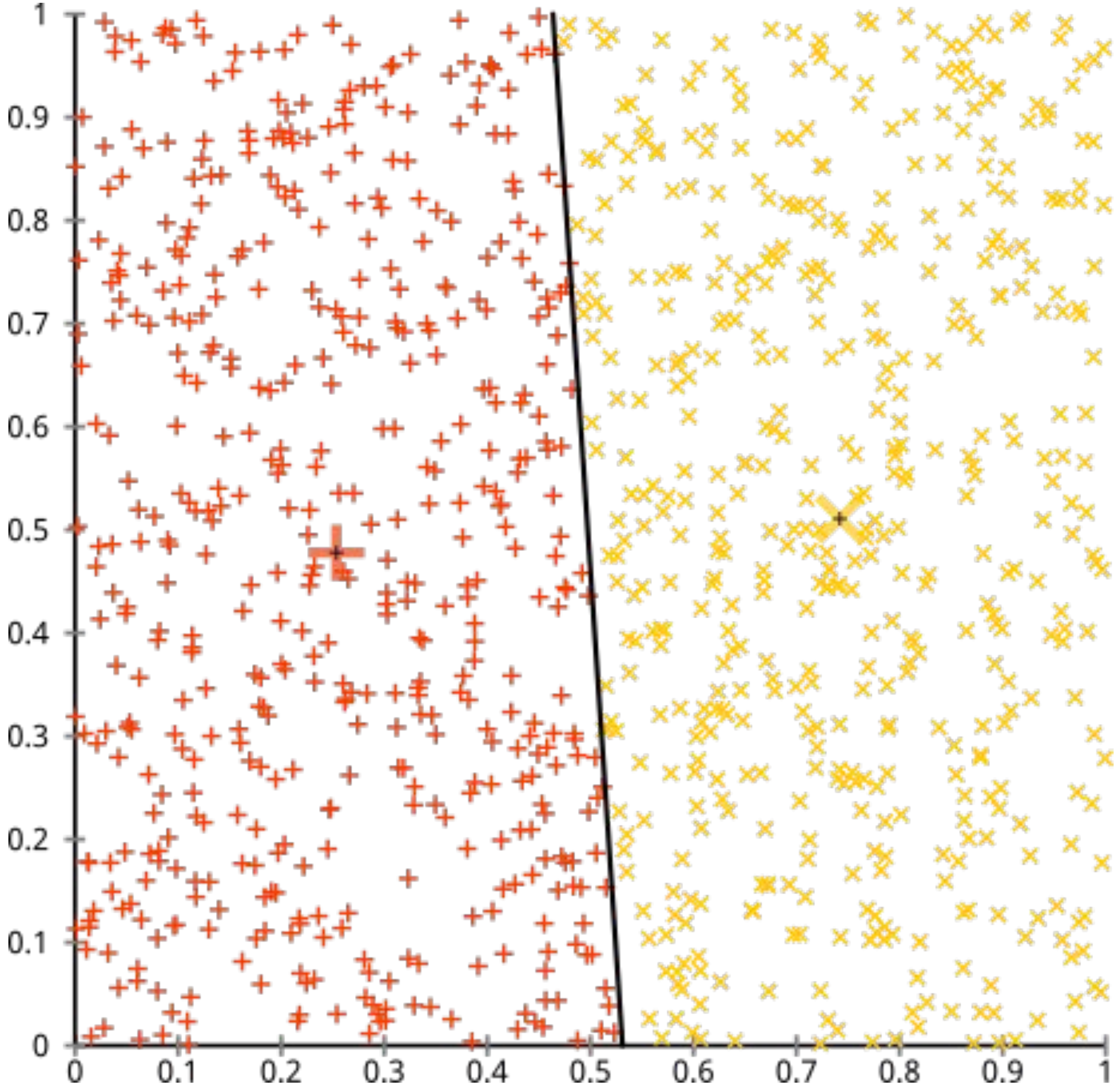
Can be improved using KMeans++



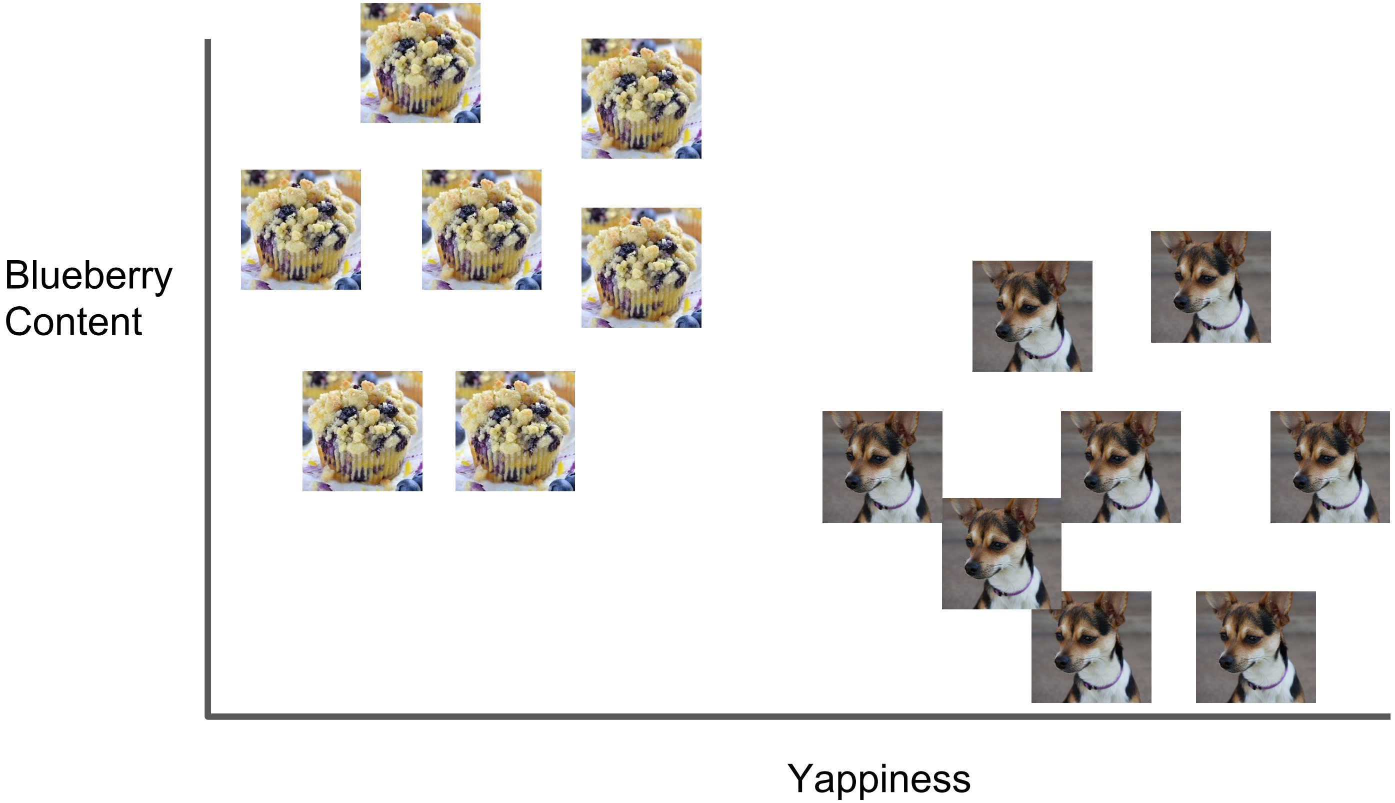
https://stats.stackexchange.com/questions/133656/how-to-understand-the-drawbacks-of-k-means?answertab=oldest#tab-top

K-Means issue #3

It will cluster non-clustered data

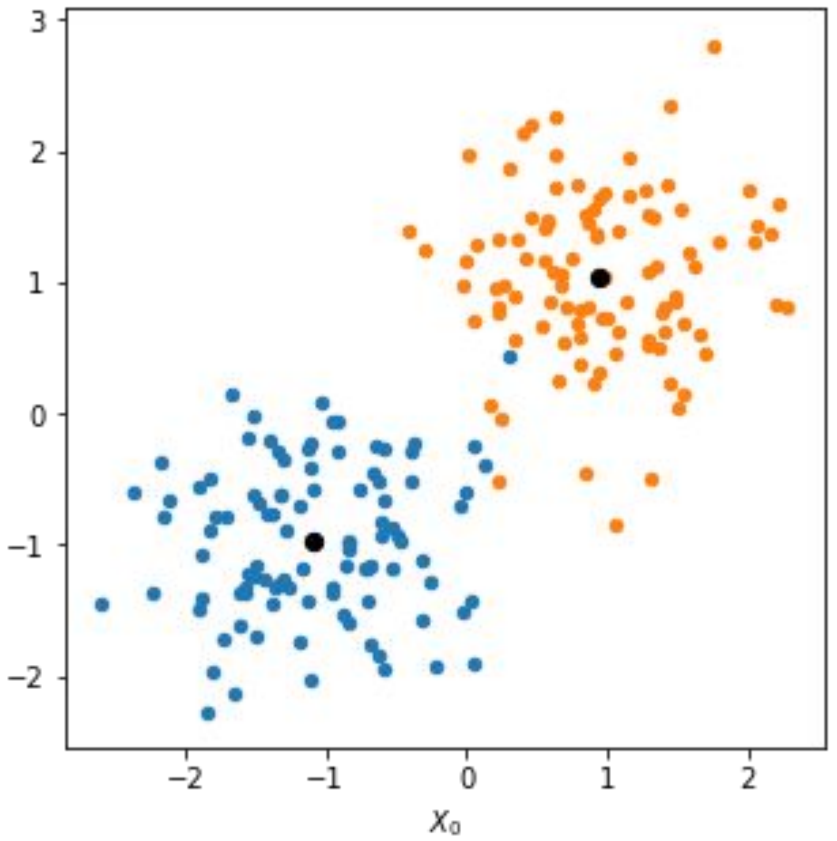


But what if we have a training set?



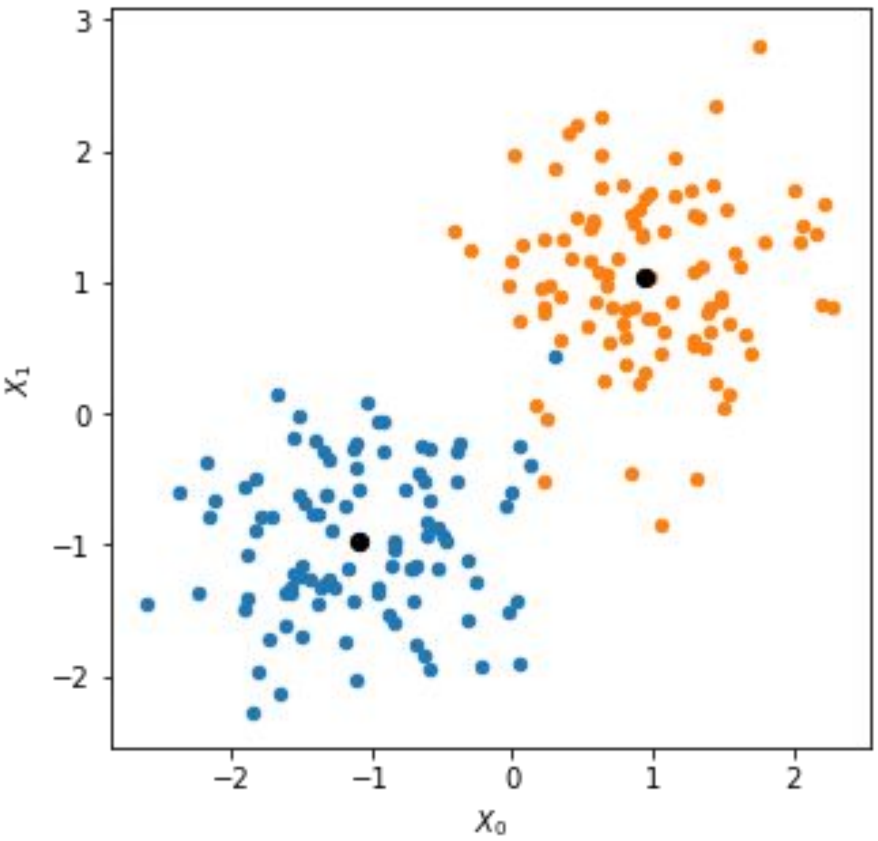
Simple approach?

- Calculate the arithmetic mean of each class
- Classify a new point by calculating the distance to each mean point and finding the minimum geometric distance

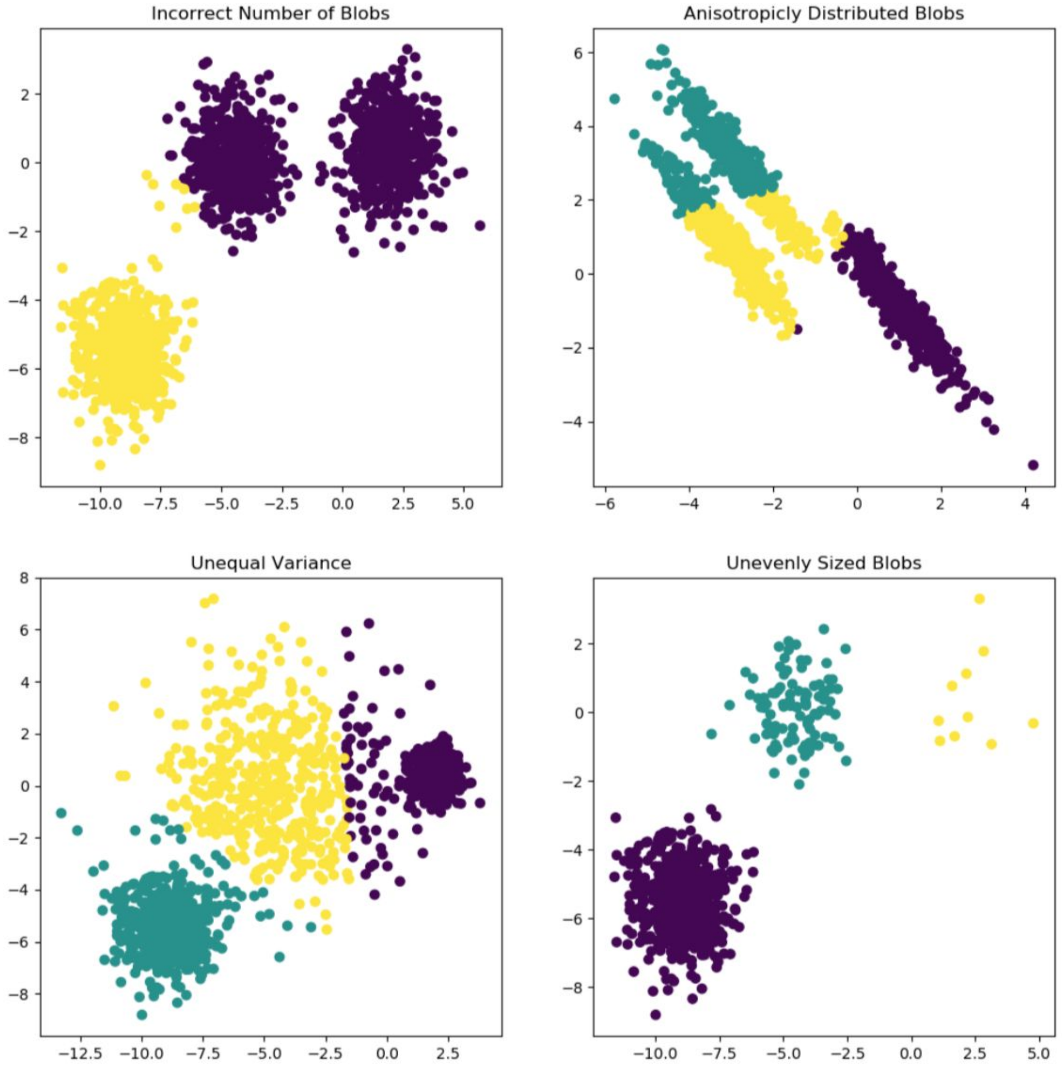


Simple approach?

- Calculate the arithmetic mean of each class
- Classify a new point by calculating the distance to each mean point and finding the minimum geometric distance
- **But this only holds if our parameters are normally distributed**

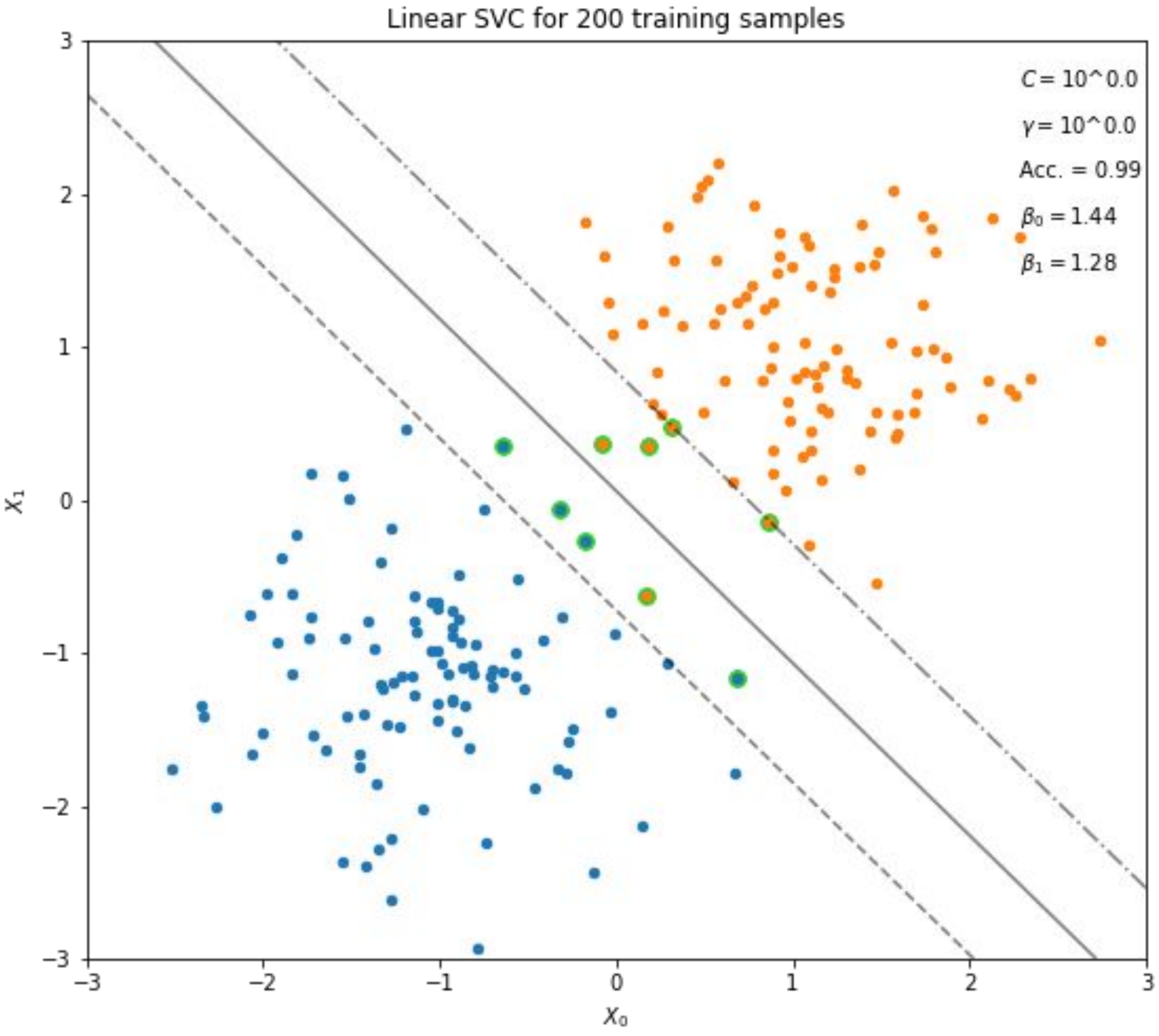


Simple approach?

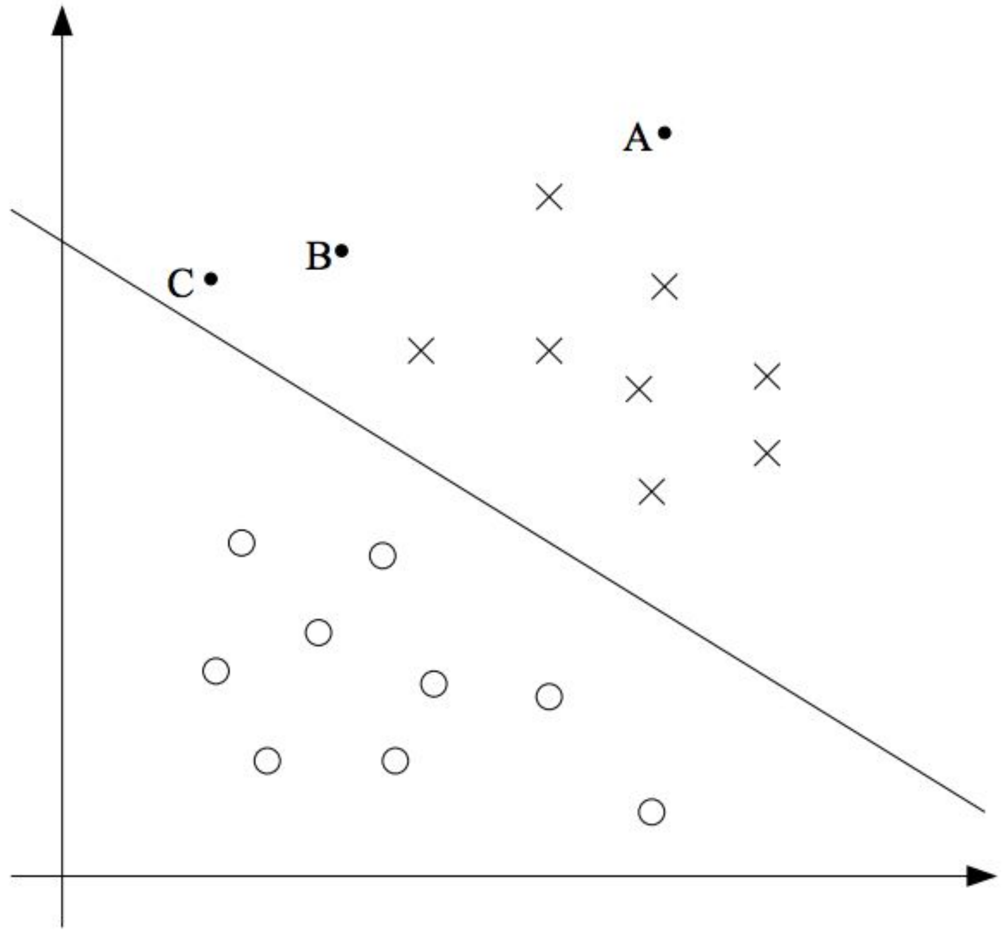


A solution?

Instead of finding the centres of the cluster, let’s focus our efforts on finding margins



A new intuition

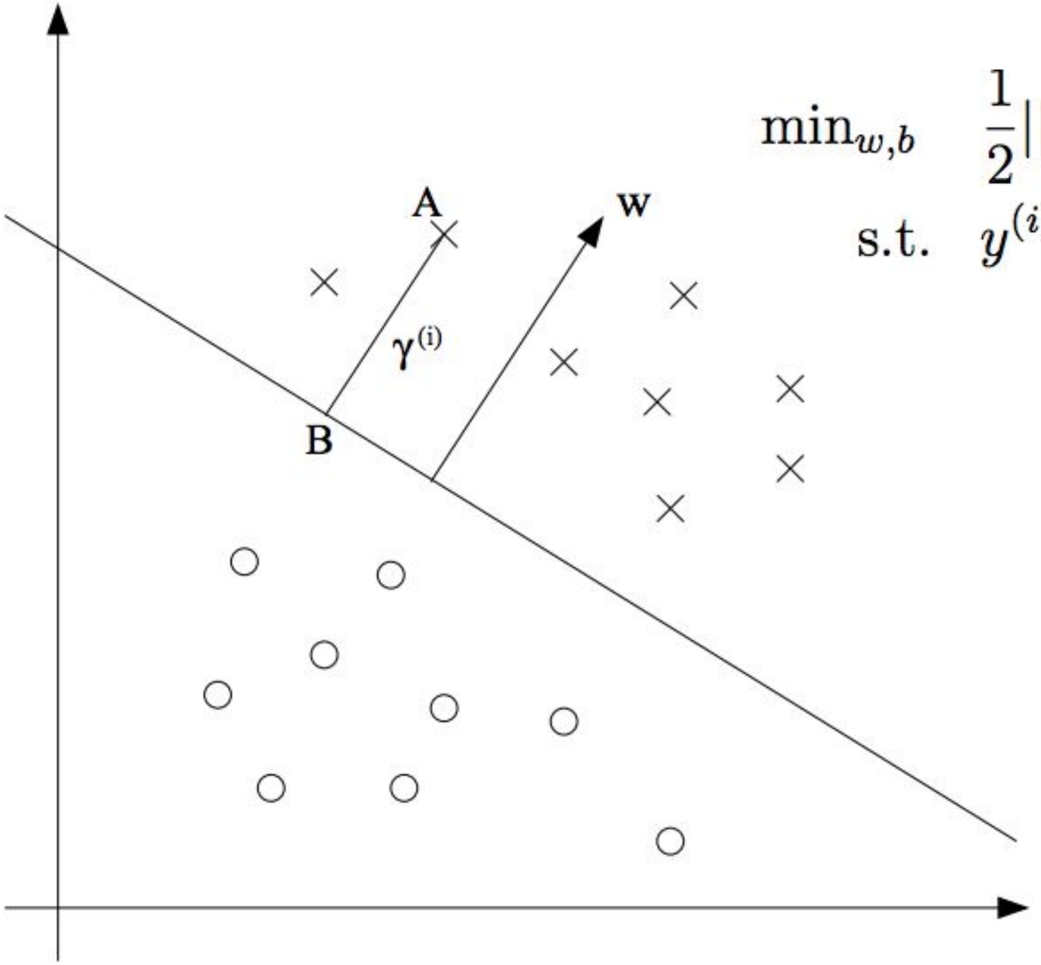


SVM

$$
\begin{aligned}
\max_{\gamma,w,b} \quad & \gamma \\
\text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \quad i = 1, \ldots, m \\
& ||w|| = 1.
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
\min_{w,b} \quad & \frac{1}{2}||w||^2 \\
\text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m
\end{aligned}
$$

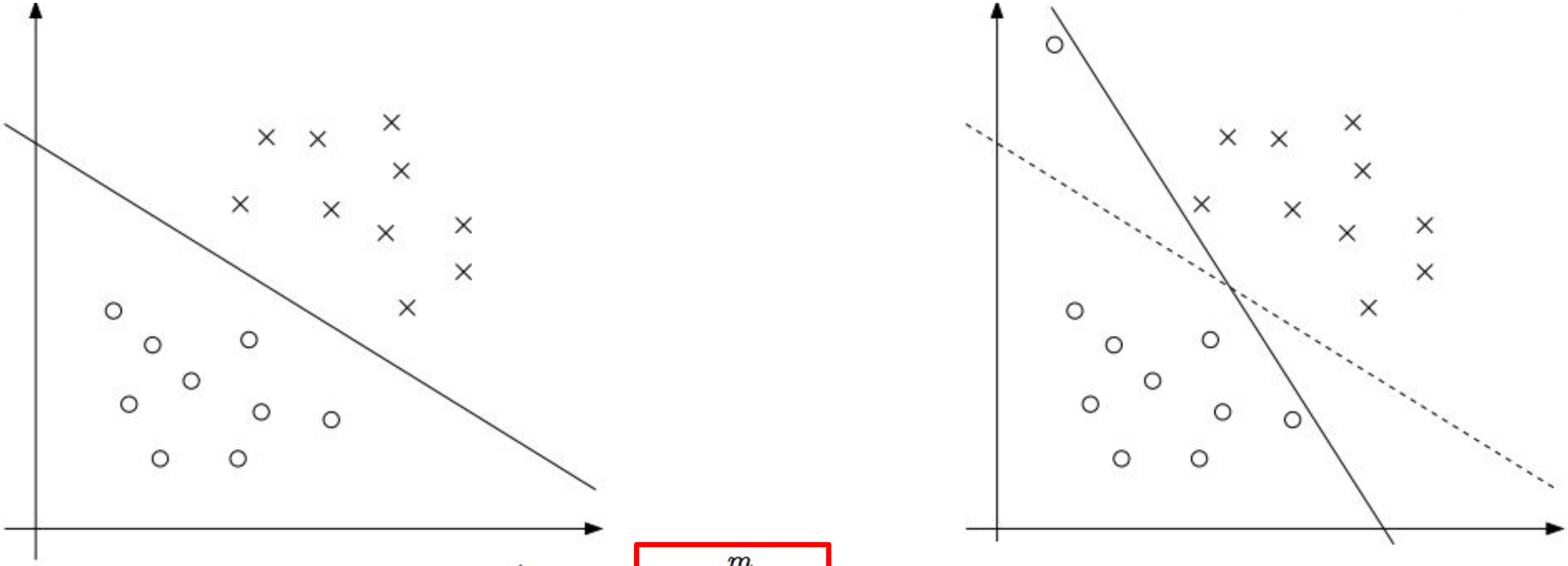


Pretty Simple?

http://cs229.stanford.edu/notes/cs229-notes3.pdf

(Full notes on SVM)

Non-separable cases



$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 \boxed{+ C\sum_{i=1}^{m}\xi_i} \\ \text{s.t.} \quad & y^{(i)}(w^Tx^{(i)}+b) \geq 1-\xi_i, \quad i=1,\ldots,m \\ & \xi_i \geq 0, \quad i=1,\ldots,m. \end{aligned}$$

The cost term

Summary points

- K-means is great for ‘well-behaved’ data in low dimensional feature space but performs badly in high dimensional feature space
- Classifying in terms of the mean is somewhat difficult with less well behaved data - other clustering techniques
- For supervised learning, SVM can be really successful but is costly to train with very large data