Basics of Machine Learning

Semester 1, 2021

Kris Ehinger

Announcements

- Please join the Ed forum using invite in your email
- Please install Jupyter Notebook before your first practical session
- If you plan to come to campus, complete the COVIDSafe module and health declaration

Outline

- Basic framework
- Levels of analysis
- Demonstration
- Data assumptions

Basic framework

Algorithms for machine learning

- Methods we'll cover:
  - Naïve Bayes classifiers
  - Decision trees
  - Support vector machines
  - Linear regression
  - Logistic regression
  - Gaussian mixture models
  - Hidden Markov models
  - Perceptron
  - Deep neural networks

Terminology

- The input to a machine learning system consists of **instances**
  - Also called **exemplars** or **observations**
  - Individual, independent samples of the world
- Instances are composed of:
  - **Attributes** (or **features**): measured aspects of each instance
  - **Concepts**: things we aim to learn (often in the form of **labels**)

Generalisation

- Learn a function that maps **attributes** to **concepts**

  concept = f(attributes)

- **Generalisation:** return the concept for any set of attributes, even ones the model has never seen before

Example: weather dataset

Outlook	Temperature	Humidity	Windy	Play?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
…	…	…	…	…

Example: weather dataset

Outlook	Temperature	Humidity	Windy	Play?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
...	...	...	...	...

Instance

Example: weather dataset

Outlook	Temperature	Humidity	Windy	Play?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
…	…	…	…	…

Attribute

Example: weather dataset

Outlook	Temperature	Humidity	Windy	Play?
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
...	...	...	...	...

Concept

Concepts

- Concepts are anything you learn
  - Discrete class labels (classification)
  - Numeric output (regression)
  - Clusters
  - Probability of an event
  - The most likely order of events
  - A sequence of commands
  - A complex model
  - …

Supervised vs. unsupervised

- **Supervised** methods receive labelled instances during training, learn the associations between attributes and concepts
- **Unsupervised** methods receive unlabelled data and learn from attributes only:
  - Discover structure in a dataset (correlated features, groups, sequences, etc.)

    潜在的
  - Discover **latent variables** that explain patterns in the observed instances
  - **Reduce dimensionality** for a supervised learner

Supervised train & test

- Goal: learn mapping from attributes to concepts

  concept = f(attributes)

- **Training:** model sees many examples of attributes-concepts pairs
- Model learns a function f() to relate them
- **Test:** model sees a new set of attributes, predicts concept
- **Evaluation:** compare prediction to ground truth

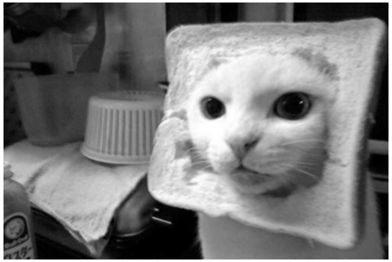
Supervised learning: Train

“yes”

“no”

Supervised learning: Test

- What is the label?



1

Model: “yes”

2

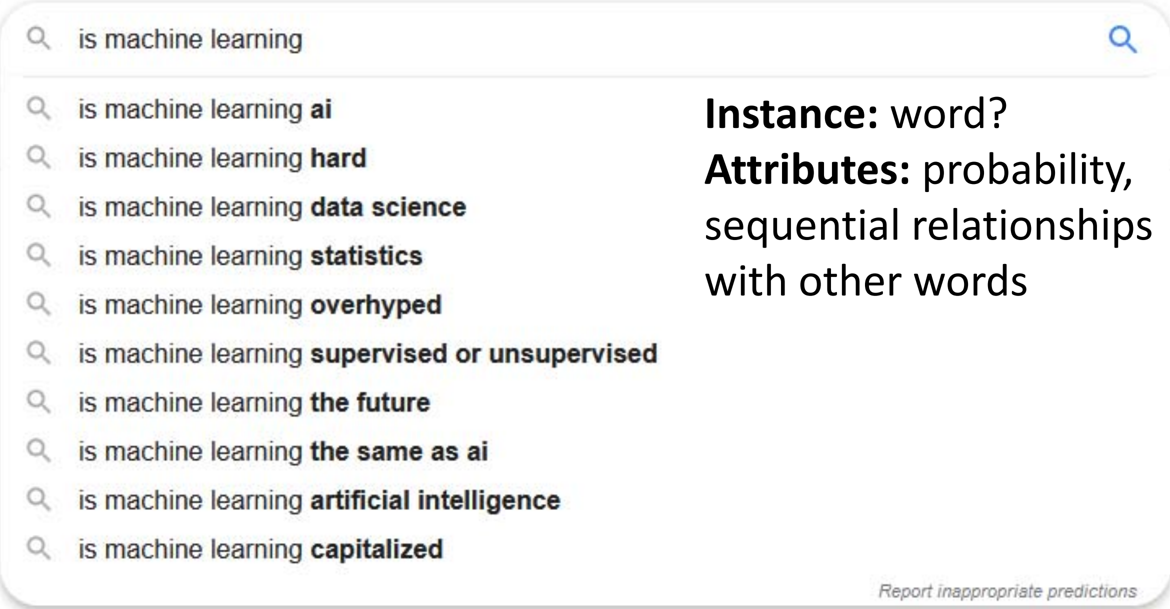
“no”

3

“yes”

Images: COREL Photos; https://www.reddit.com/r/funny/comments/j6lyw/if_i_just_keep_still_someone_is_bound_to_put/; https://www.themarysue.com/things-that-look-like-faces-pareidolia/

Unsupervised example



Instance: word?
Attributes: probability, sequential relationships with other words

Unsupervised train & test

- Goal: learn mapping from attributes to concepts

  concept = f(attributes)
- **Training:** model sees many examples of attributes
- Model learns a function f() that produces a useful concept (e.g., probability distribution)
- **Test:** model sees a new set of attributes, predicts concept
- **Evaluation:** ?
  - Probability models: see if future samples from the same distribution are well-predicted by your model

Association learning

- Detect useful patterns, associations, correlations or causal relations between attributes or between attributes and concept
- A good pattern is a combination of attribute values where the presence of certain values strongly predicts the presence of other values
- Any kind of structure is considered interesting and there may be no “right” answer
- Evaluation can be difficult, potentially many possible association rules in one dataset

Association learning example

```
# java weka.associations.Apriori -t data/weather.nominal.arff
```

```
 1. humidity=normal windy=FALSE ==> play=yes
 2. temperature=cool ==> humidity=normal
 3. outlook=overcast ==> play=yes
 4. temperature=cool play=yes ==> humidity=normal
 5. outlook=rainy windy=FALSE ==> play=yes
 6. outlook=rainy play=yes ==> windy=FALSE
 7. outlook=sunny humidity=high ==> play=no
 8. outlook=sunny play=no ==> humidity=high
 9. temperature=cool windy=FALSE ==> humidity=normal play=yes
10. temperature=cool humidity=normal windy=FALSE ==> play=yes
```

Levels of analysis

Marr’s levels of analysis

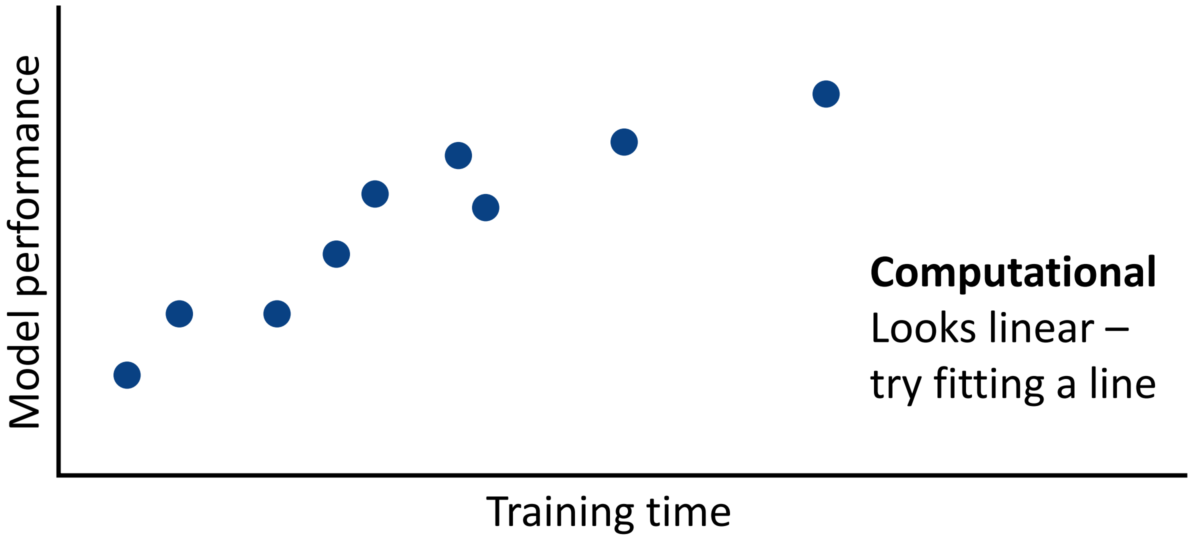
- Framework for understanding information-processing systems
- **Computational** level
  - What is the goal of this system?
- **Algorithmic** level
  - How do you achieve the goal?
  - Algorithms and data structures
- **Implementational** level
  - Physical implementation (circuits, neurons)

Marr (1982)

Machine learning framework

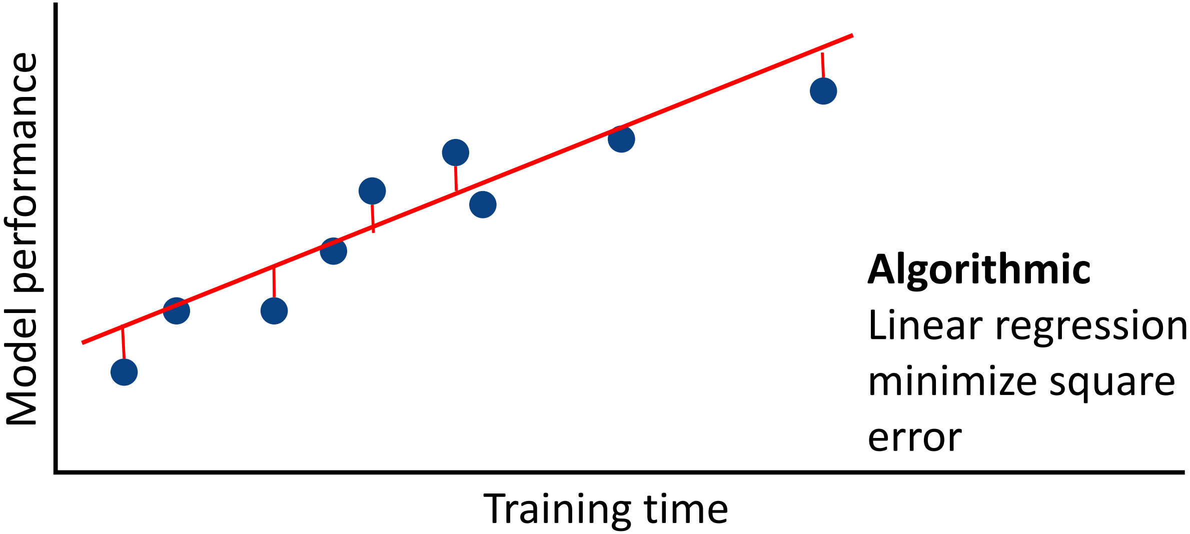
- **Computational** level
  - What structure does this machine learning model expect to see in the world?
  - What rule/pattern/model/etc. explains this data?
- **Algorithmic** level
  - Given a model, what's the best fit for this data?
  - Usually involves minimizing an error or loss function
- **Implementational** level
  - How to find that best fit in finite time?
  - Not always possible to solve exactly

Example: linear regression



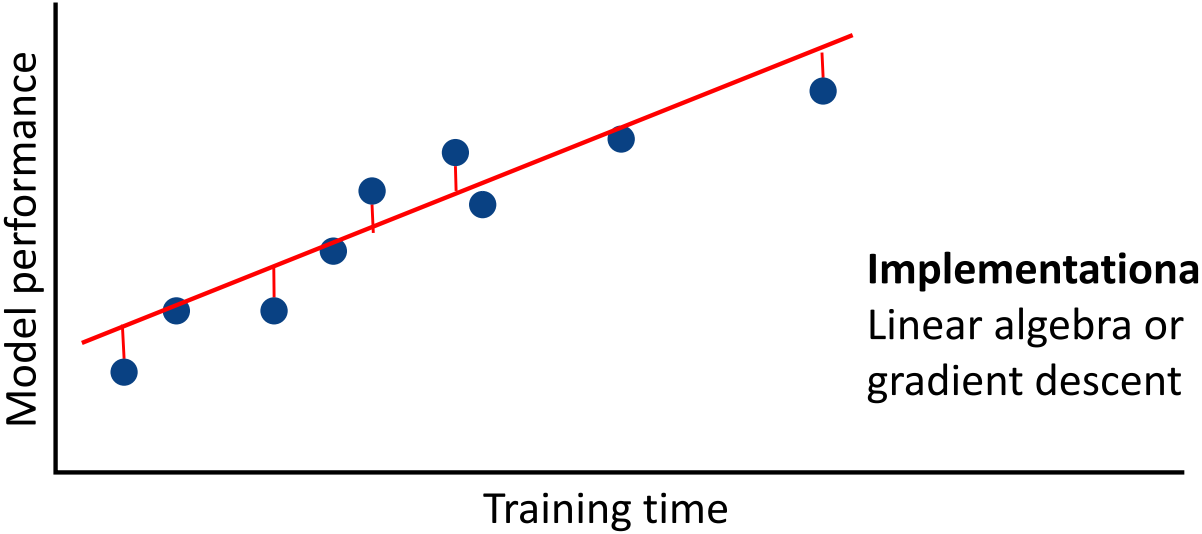
Computational
Looks linear –
try fitting a line

Example: linear regression



Algorithmic
Linear regression, minimize square error

Example: linear regression

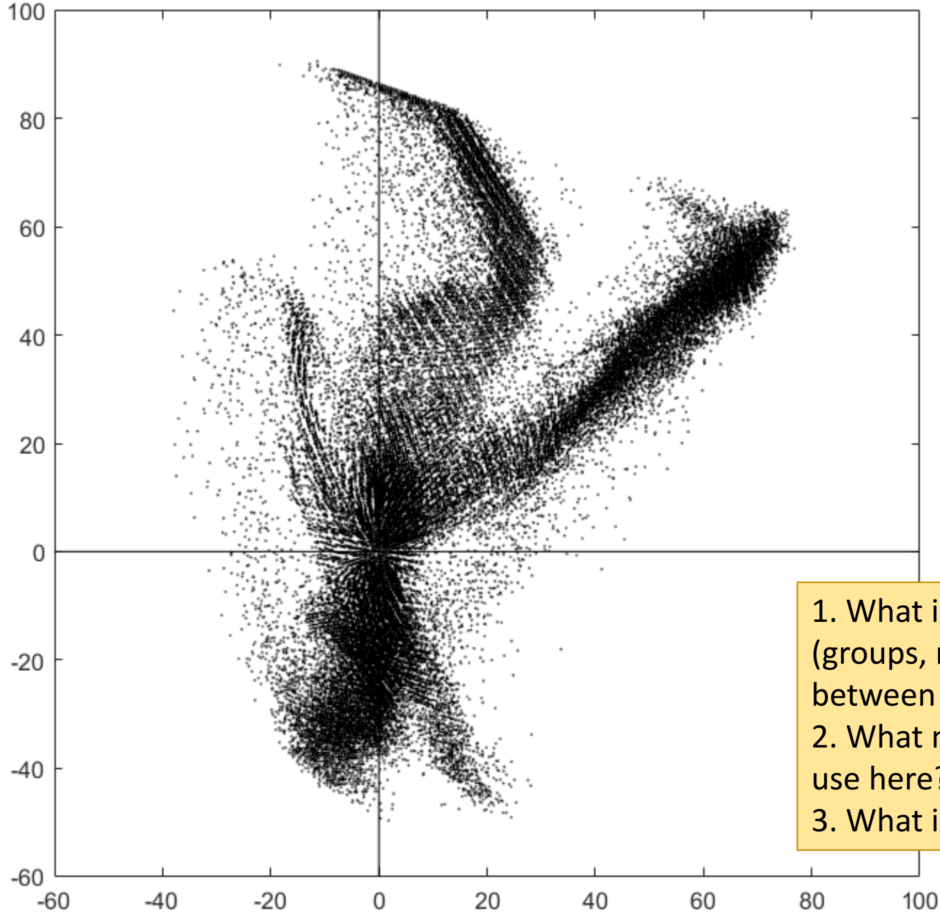


Implementational
Linear algebra or
gradient descent

Demonstration

Example problem

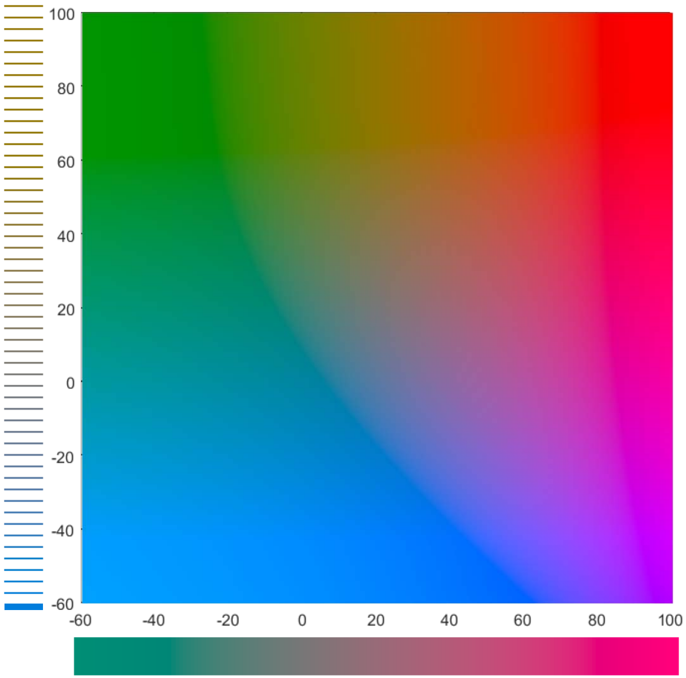
- What structure do you see in this data?
  - Are there groups? How many?
  - Are there differences between the groups?
  - Any other correlations / relationships?
- What machine learning algorithm might you use to model this data?
  - Hint: It's an *unsupervised* learning problem
- Bonus question: What is this data?

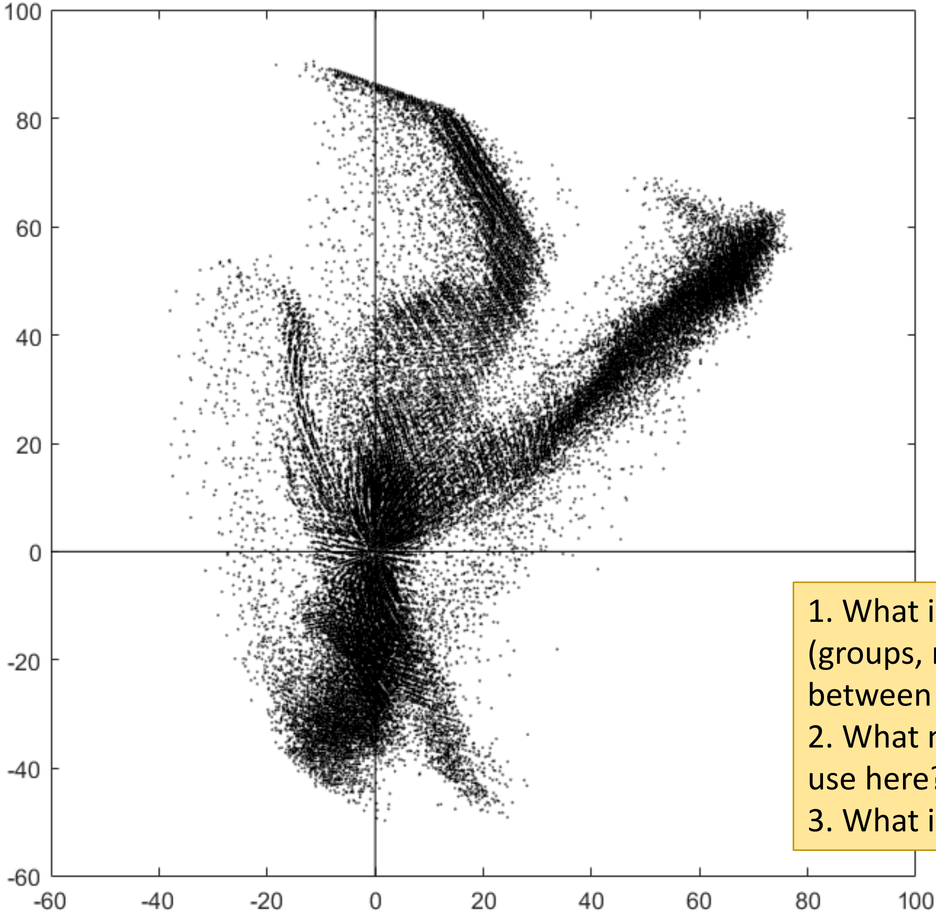


250,000 instances
attributes: [x y]

1. What is the structure? (groups, relationships between x & y)
2. What model would you use here?
3. What is this data?

Colour histogram

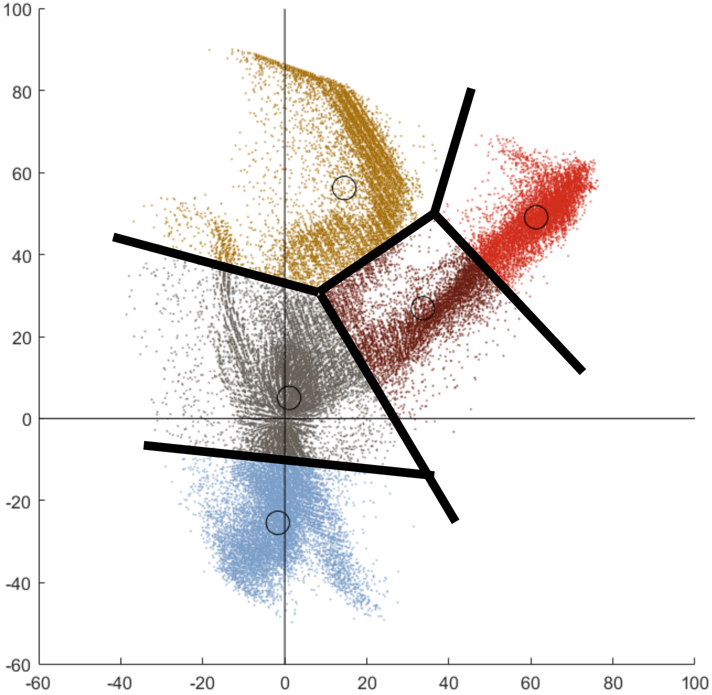




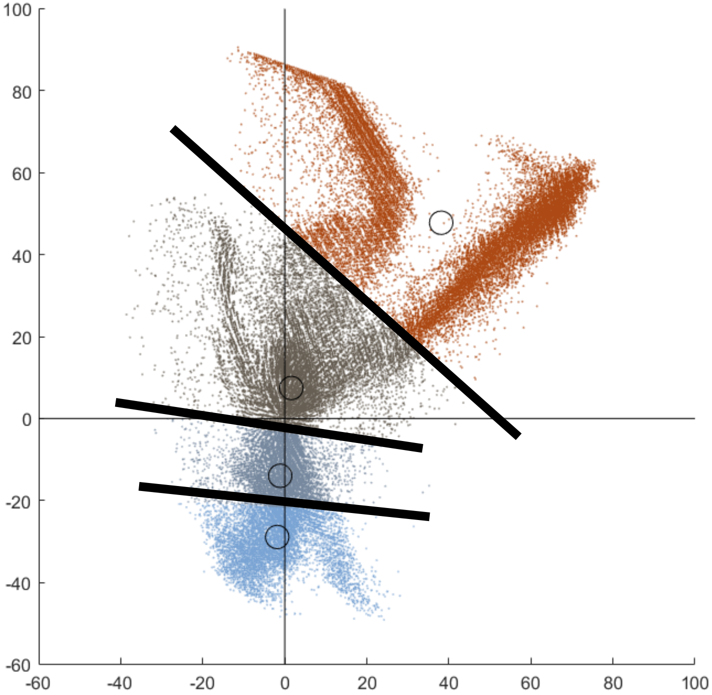
250,000 instances
attributes: [x y]

1. What is the structure? (groups, relationships between x & y)
2. What model would you use here?
3. What is this data?

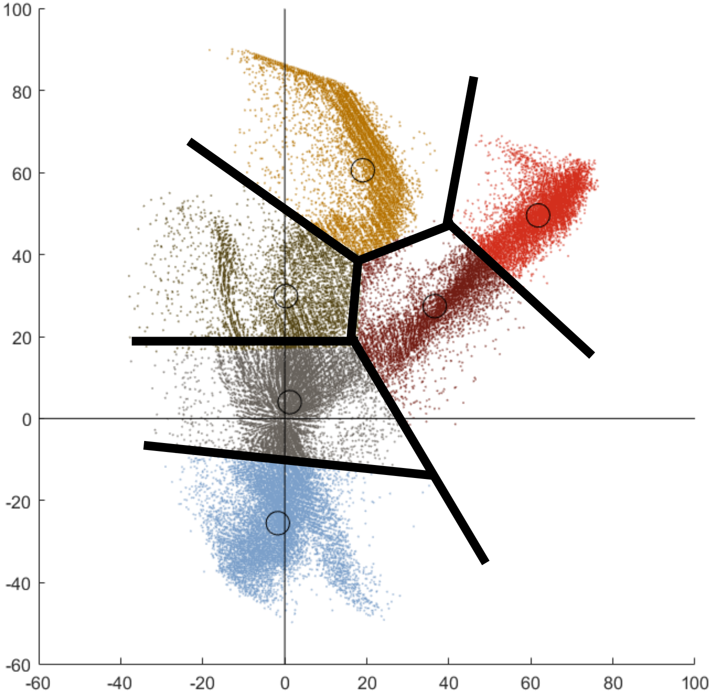
K-means (5 clusters)



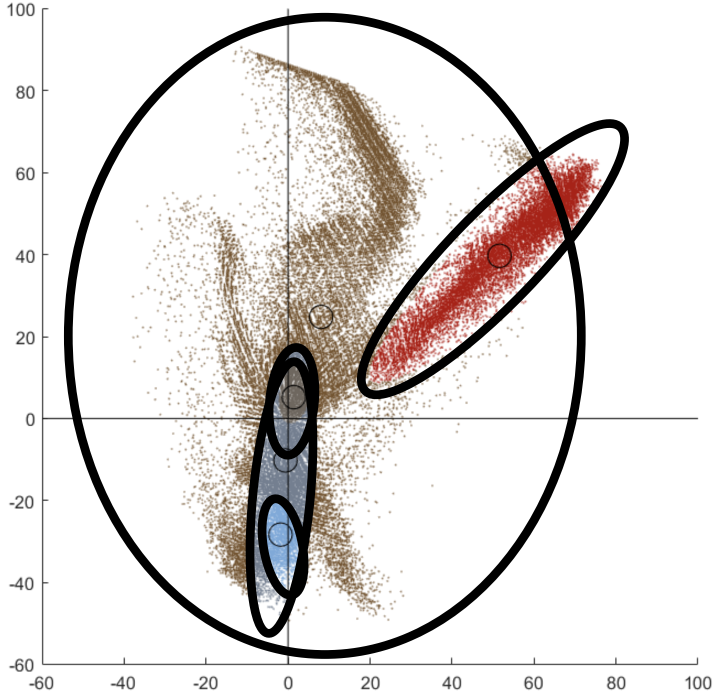
K-means (4 clusters)



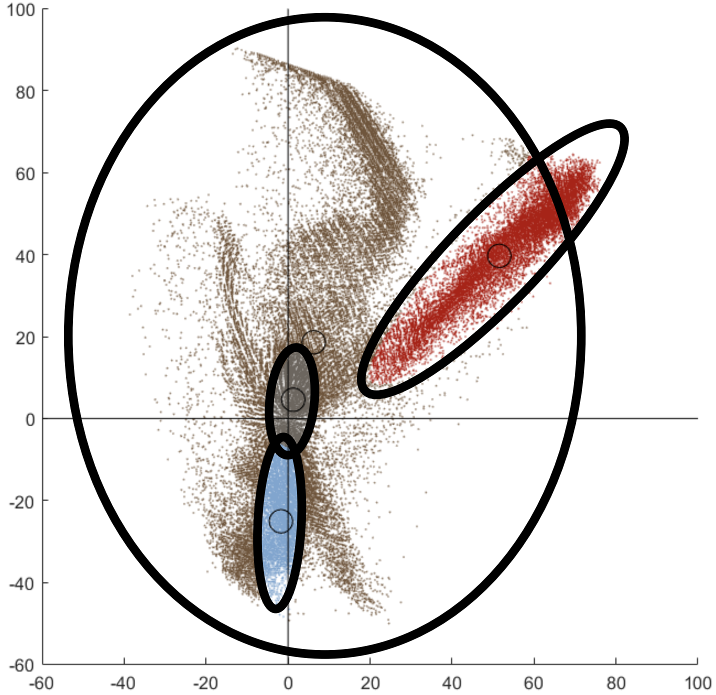
K-means (6 clusters)



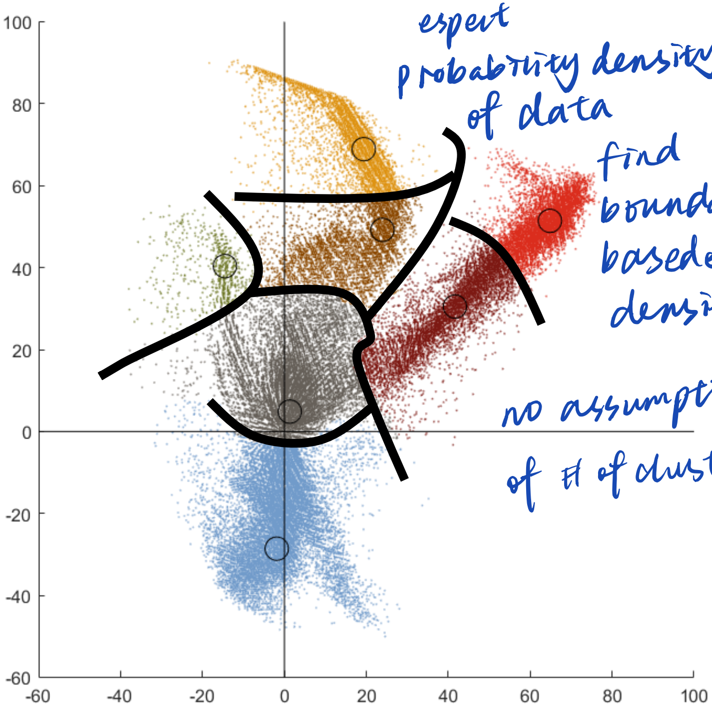
Gaussian mixture model (5)



Gaussian mixture model (5)



Mean shift

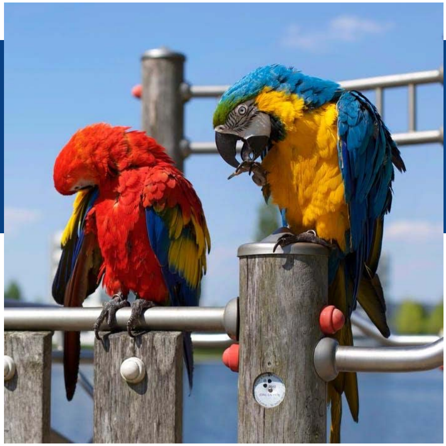


espect probability density of data

find boundary based on density

no assumption of # of clusters

Which is correct?



5-cluster solutions:

K-means

Gaussian mixture model

Mean shift

Summary

- Even when models have the same goal (find clusters) they make very different assumptions which leads to different results
- Fewer assumptions = better model?
  - Not necessarily! Models that make some assumptions to simplify the problem may find a better result