WEEK 9: ANOMALY DETECTION

STK-INF 3000/4000

DIRK HESSE

ANOMALY DETECTION

EXAMPLES

- Detect if a credit card was stolen.
- Detect if a account has been hacked.
- Detect if a piece of equipment functions normally.
    - Predicting hard disk failures.
    - Predicting plane engines about to break.
    - Detect production errors.
- Detect if a insurance claim is fraudulent.

TYPES OF ANOMALY DETECTION.

- Expert systems.
- Statistical anomaly detection.
- Network analysis.

EXPERT SYSTEMS

- A lot of *if* - *then* rules.
  - If the vibration of an engine increases and the temperature sinks, ring an alarm.
  - If a stock price went down by more than a standard deviation in a week, short it.
- Used a lot in the past.

WHY RULES FAIL...

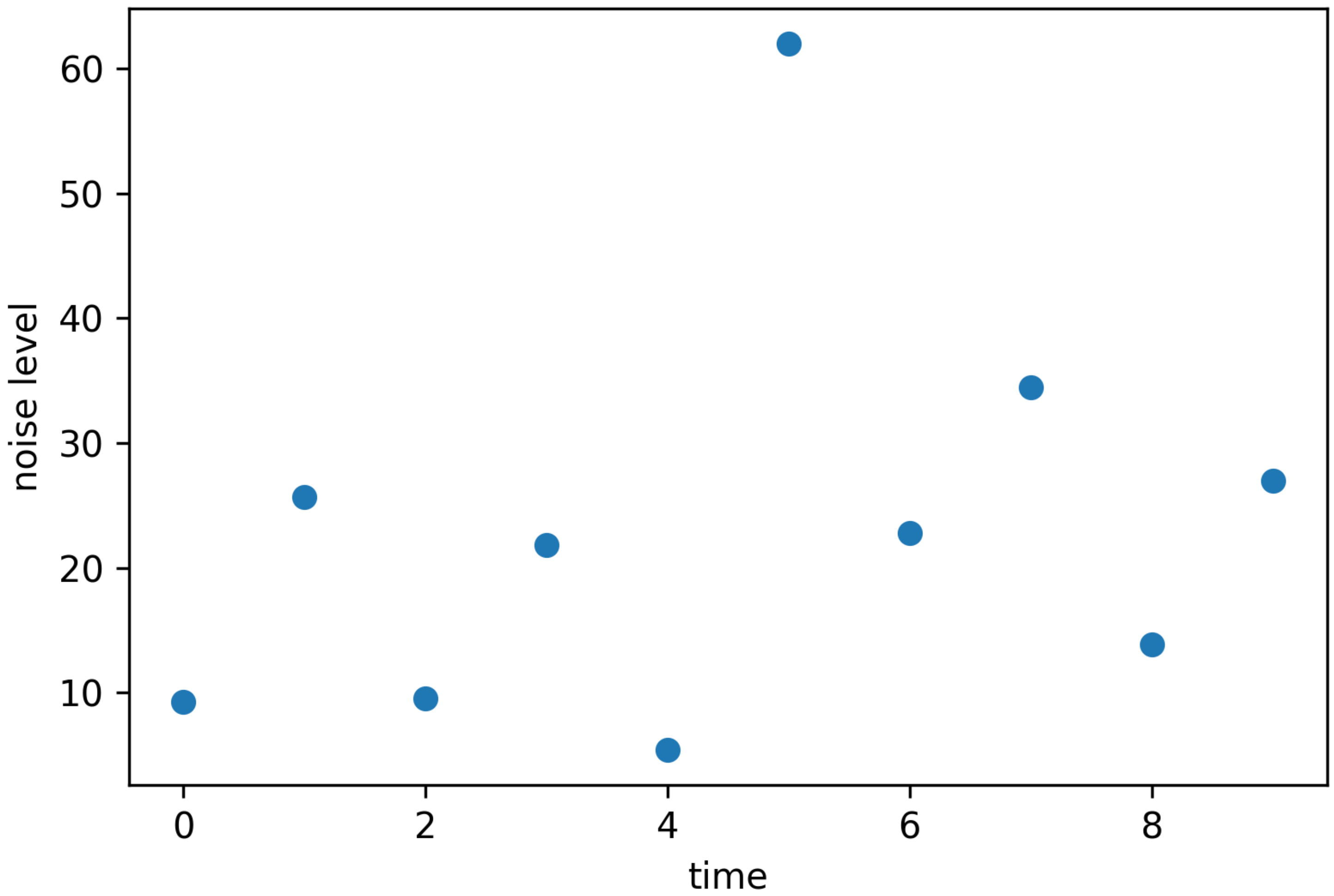
- Rigid.
- Hard to maintain.
- Hard to explain.
- Will only find what you're looking for.

ANOMALY DETECTION

UNSUPERVISED LEARNING

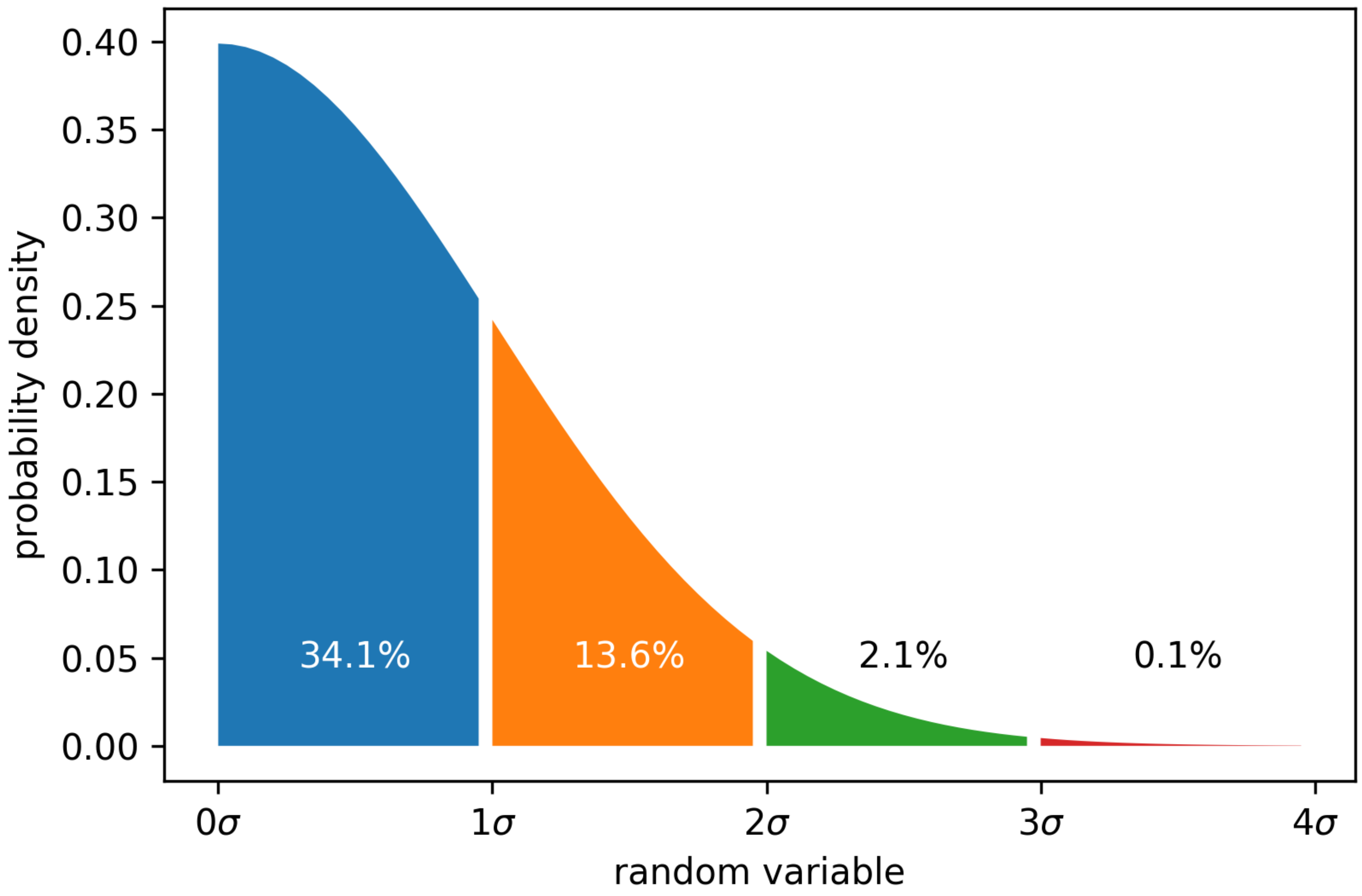
- Anomaly detection uses (usually) unsupervised learning.
- Have a bunch of x_i, no target.
  - Cases of fraud might be unknown.
  - Or too rare to make a good predictor.
- Try to make sense of the x_i.
  - Usually means finding an approximation of $p(X = x)$ given the training data.
  - Gives a measure how improbable the observation is.

EXAMPLE: PREDICTING MACHINE MALFUNCTION.



60
50
40
30
20
10
noise level
0
2
4
6
8
time

THE NORMAL DISTRIBUTION


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
probability density
34.1%
13.6%
2.1%
0.1%
0σ
1σ
2σ
3σ
4σ
random variable

THE Z VALUE

- Assume we have data $x_1, \ldots, x_N$.
- Calculate the mean $\overline{x} = \frac{1}{N} \sum_i x_i$.
- Calculate the standard deviation $\sigma = \sqrt{\frac{1}{N} \sum_i (x_i - \overline{x})^2}$
- Calculate the z-value $z_i = (x_i - \overline{x})/\sigma$.
- Flag everything with $z > z_{\mathrm{max}}$ as anomaly.

CHEBYSHEV'S INEQUALITY

- Valid for a wide variety of *probability distributions*.
- Statement:

$$Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

- I.e. looking at z values for anomaly detection makes sense.

SIGMAS AND PROBABILITIES

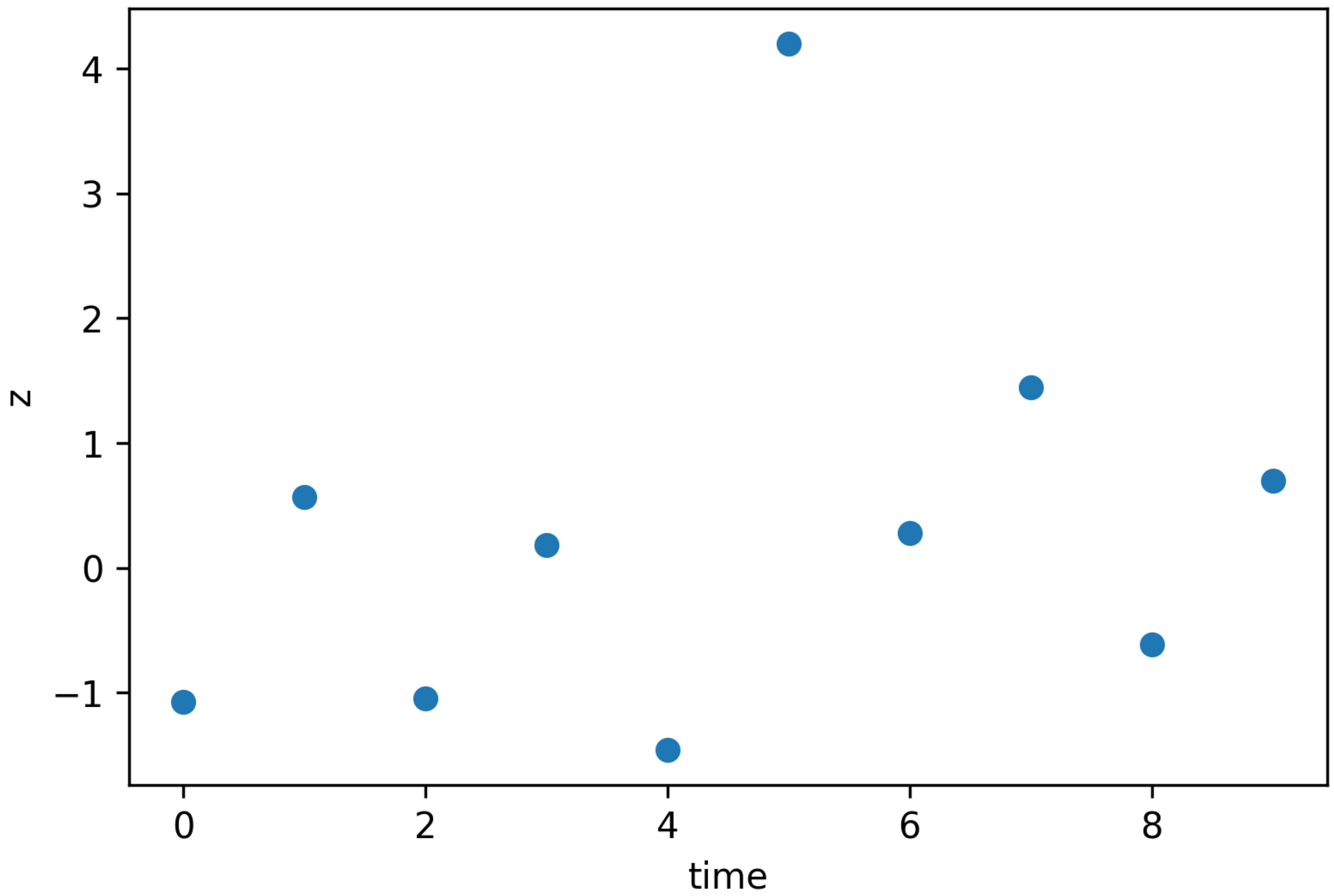
For the normal distribution:

Threshold	Fraction Outside
3σ	1 / 370
4σ	1 / 15 787
5σ	1 / 1 744 278
6σ	1 / 506 797 346

CHEBYSHEV GUARANTEES

Threshold	Percent Outside
3σ	11.1111%
4σ	6.25%
5σ	4%
6σ	2.7778%

EXAMPLE: PREDICTING MACHINE MALFUNCTION.



4
3
2
1
0
−1
z
0
2
4
6
8
time

MULTIDIMENSIONAL DATA

Can fit a multivariate normal distribution

$$f(x) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^d|\Sigma|}}$$

$$\mu = \frac{1}{N}\sum_i x_i$$

$$\Sigma = \frac{1}{N-1}\sum_i (x_i-\mu)(x_i-\mu)^T.$$

Flag everything with $f(x) < \epsilon$ as anomaly.

THE NEED FOR CLUSTERING

- There might be natural variations in data.
  - Weekend vs. weekday spending patterns.
  - Heart rhythms at rest vs. during sport.
- Fit a bunch of (multivariate) normal distributions.
  - How? We generally don't have labels.
- Enter: Cluster methods.

CLUSTERING

CLUSTERING 101

- Needed: Some measure for distance between measurements.
  - I.e. a metric.
- Objective: Find k clusters of points that are close together.
  - Some methods find k automatically.
  - Most methods need it as input.

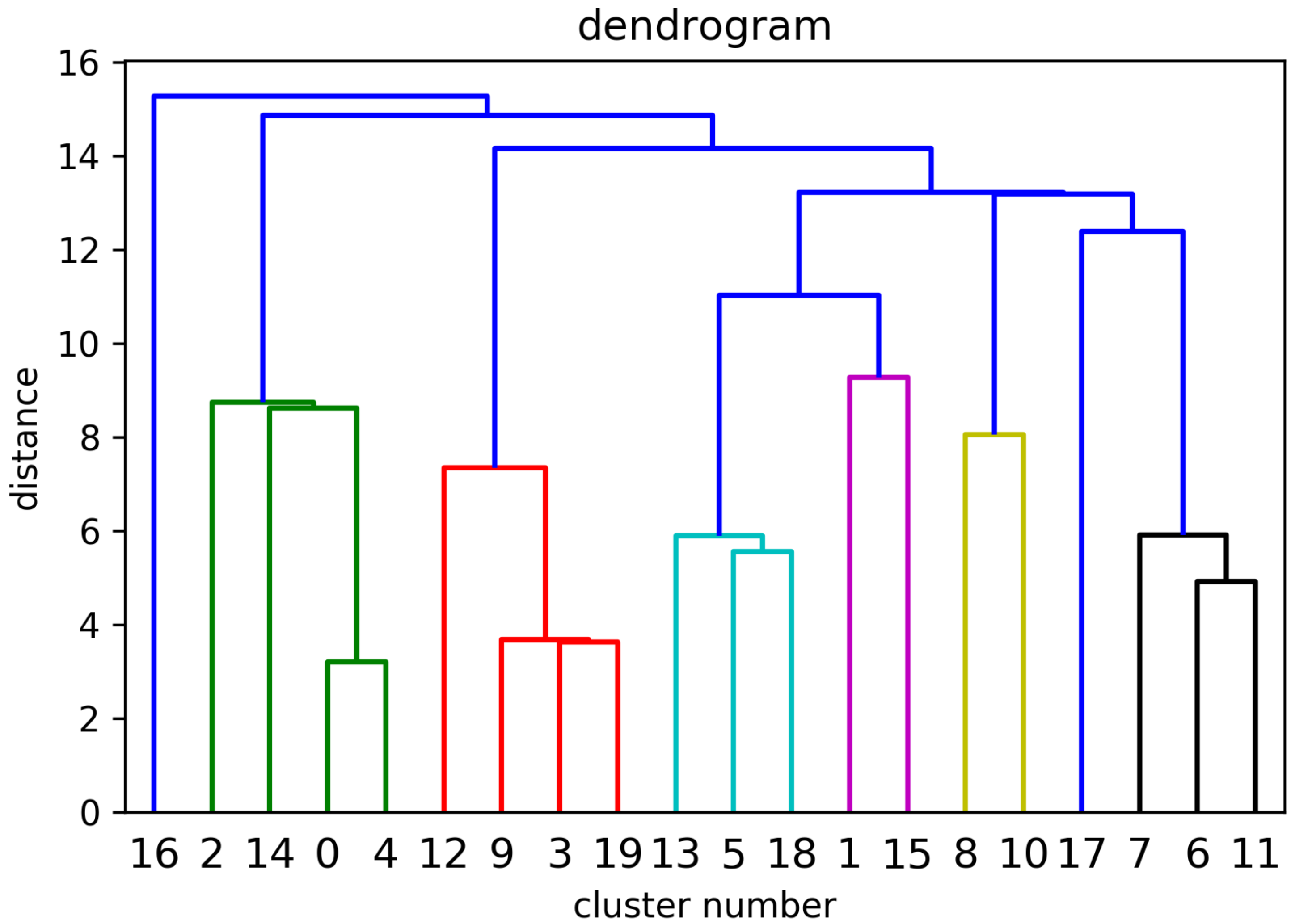
TYPES OF CLUSTERING

- Hierarchical
  - Arranges samples in a hierarchy, according to distance.
  - Types:
    - Agglomerative (bottom-up)
    - Divisive (top-down, not terribly common)
- Non-hierarchical
  - Examples:
    - K-Means.
    - DBSCAN.

AGGLOMERATIVE CLUSTERING

- Start with each x_i defining its own cluster.
- At each step, merge the closest two clusters.
  - Closest according to linkage you've chosen.
- Keep going until you have only one cluster.
- Choose a place to 'cut' the resulting tree.

AGGLOMERATIVE CLUSTERING



dendrogram
distance
16
14
12
10
8
6
4
2
0
16 2 14 0 4 12 9 3 19 13 5 18 1 15 8 10 17 7 6 11
cluster number

DISTANCE BETWEEN TWO CLUSTERS?

- Linkage
  - Single (closest points of clusters).
  - Full (furthest points of clusters).
  - Average (average distance of points).
  - Centroid (distance of cluster centers).

HOW TO FIND THE NUMBER OF CLUSTERS?

- Sometimes given.
  - Want to distribute customers to k service people.
- Sometimes can be 'eyeballed' by looking at the dendrogram.
- Sometimes we need to work a little.
  - Number of classes might be unknown.
  - Might be questionable if there is structure at all.

PREPARATIONS FOR FINDING K NUMERICALLY.

- Given clustering with k cluster centers μ_l.
- Assign each x_i a cluster $C(x_i)$, such that

$$C(x_i) = \operatorname*{argmin}_l \|x_i - \mu_l\|$$

- For agglomerative clustering we'll often have the $C(x_i)$ first and calculate

$$\mu_l = \frac{1}{N_l} \sum_{i;C(x_i)=l} x_i$$

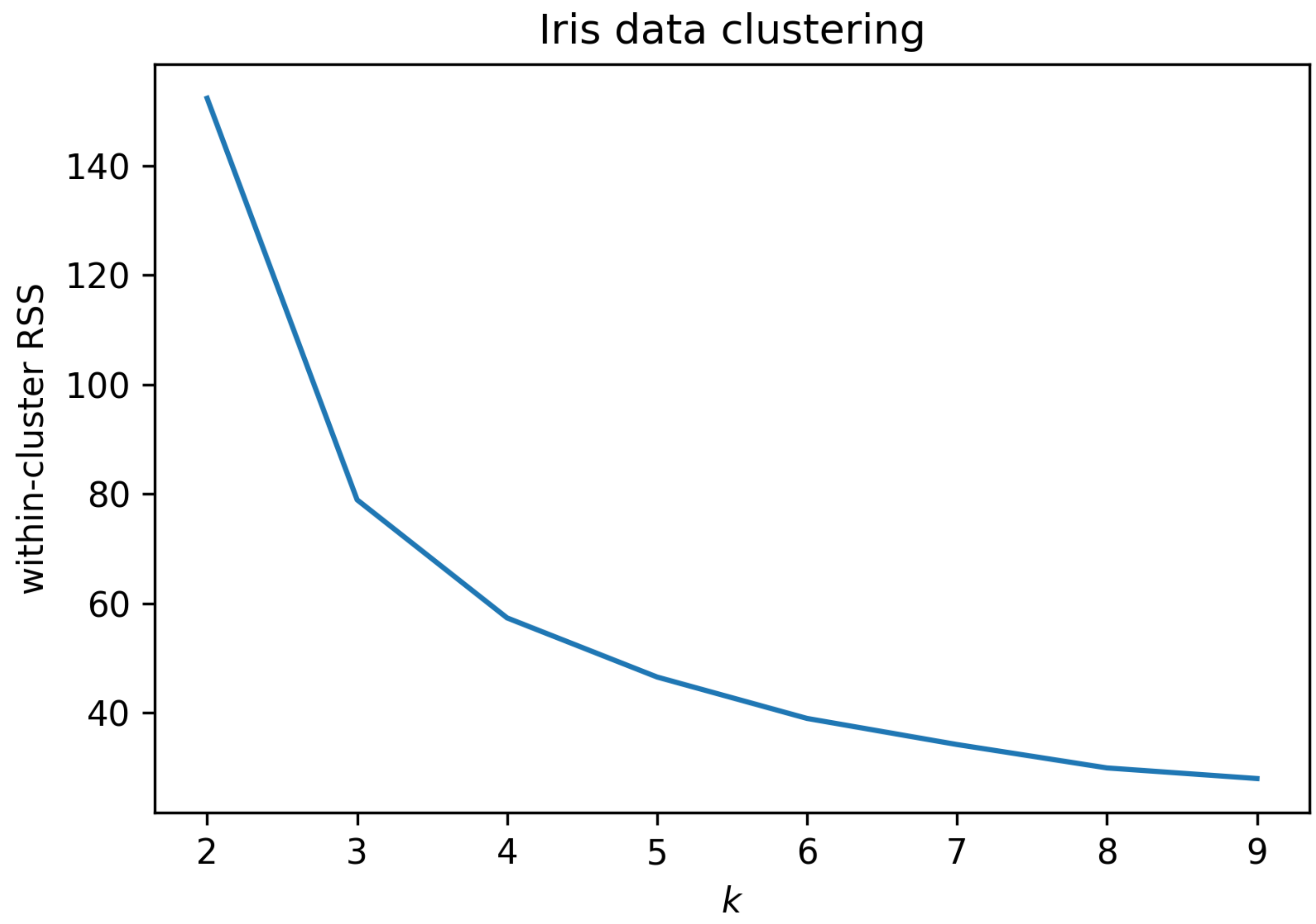
- Define the inertia (or within-cluster RSS)

$$\sum_{l} \sum_{i; C(x_i)=l} \| \mu_l - x_i \|^2$$

FINDING K NUMERICALLY.

- Set a range k you want to explore.
- Calculate the inertia for each of them.
- Plot vs. k.
- Often you'll see a knee-shape.
  - Less clusters than optimal: High reduction in variance.
  - More clusters than optimal: Don't gain much.
- Choose k at or close to 'knee'.

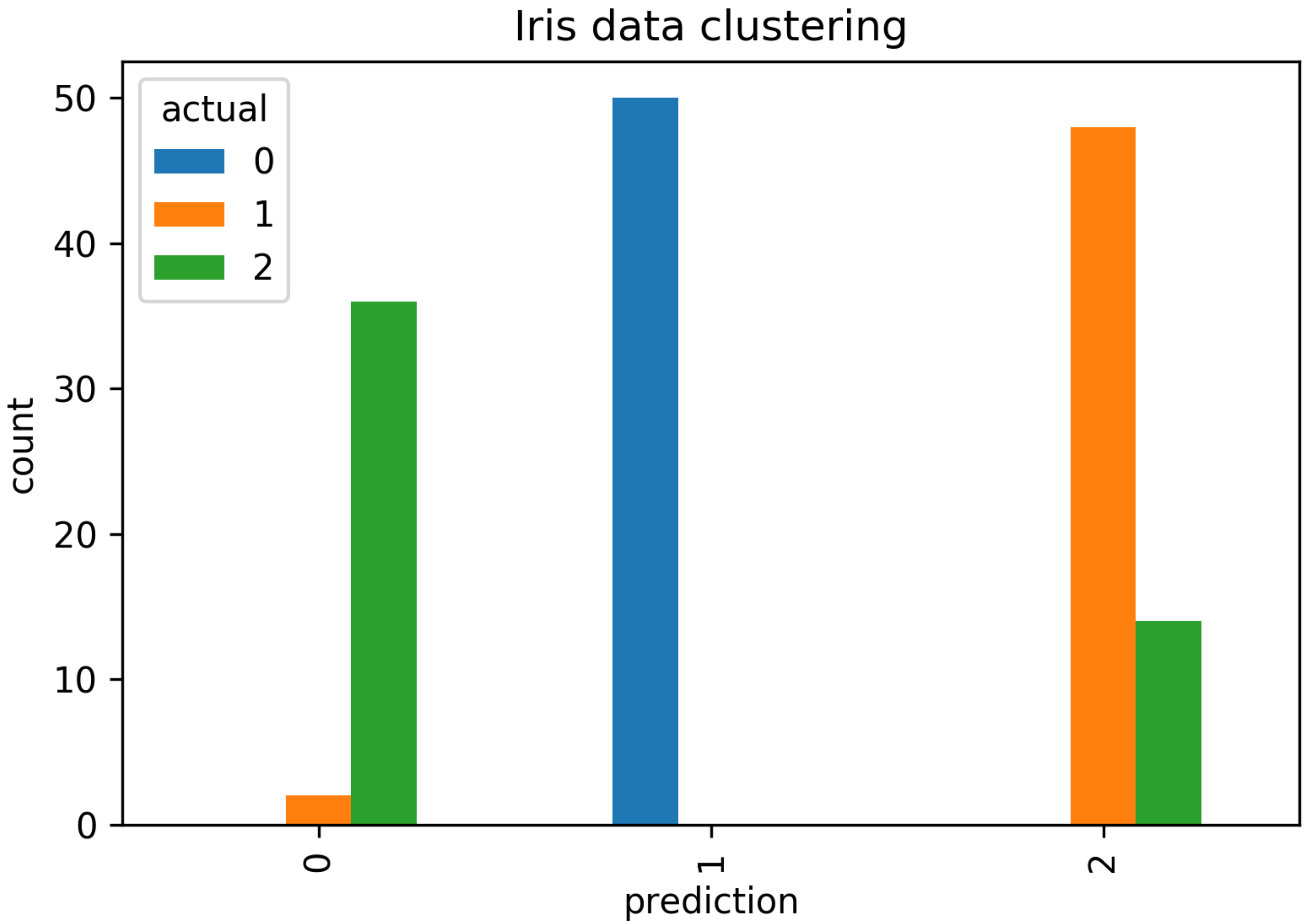
K-SCAN FOR THE IRIS DATA SET.


Iris data clustering
within-cluster RSS
140
120
100
80
60
40
2
3
4
5
6
7
8
9
k

HOW DO YOU KNOW YOU DID WELL?

- Inspect per-cluster means of $X^{(i)}$.
  - Do they separate well?
- Inspect known labels.
  - Do we classify correctly?
  - Beware of cluster label mismatch.

EVALUATING THE IRIS CLUSTERS.


Iris data clustering
actual
0
1
2
50
40
30
20
10
0
count
0
1
2
prediction

K-MEANS CLUSTERING

- Start with k *random* cluster means μ_l.
- Given the data x_i, calculate new cluster centers, again using

$$C(x_i) = \underset{l}{\operatorname{argmin}} \|x_i - \mu_l\|$$

$$\mu_l' = \frac{1}{N_l} \sum_{i;C(x_i)=l} x_i$$

- Iterate until the assignment stops changing.

PROS AND CONS

- K-Means
  - Fast ($O(N^2)$).
  - Good for sphere shaped clusters.
  - Some randomness.
    - Starts with random center assignments.
    - Outcome might vary from run to run.
- Hierarchical clustering.
  - Slower ($O(n^2)$).
  - Works well with any cluster shape.
  - Can eyeball k from dendrogram.

PRACTICAL CONSIDERATIONS

- Needs a good metric.
- Observations might need to be standardized.
- Metric chosen might be of interest.
  - 1-Norm (aka Manhattan distance).
  - l-Norm (usually Euclidean).

CLUSTERING FOR FRAUD DETECTION

- Calculate the per-cluster standard deviation

$$\sigma_l = \sqrt{\frac{1}{N_l} \sum_{i; C(x_i) = l} \|x_i - \mu_l\|^2}.$$

- And the per-cluster z-value

$$z_i = \frac{x_i - \mu_{C(x_i)}}{\sigma_{C(x_i)}}.$$

- Classify points as an **anomaly** if

$$z_i > z_{\text{max}}.$$

QUESTIONS?