Machine Learning Introduction
MCHE 470: Robotics

Dr. Joshua Vaughan
Rougeou 225
joshua.vaughan@louisiana.edu
@Doc_Vaughan

What is Machine Learning?

- Applying some knowledge about the past to:
  - Analyze the present
  - Predict the future

Example Applications

- Classification
  - An object is red and round, is it an apple?
  - An object has **X-like properties**, is it ____**X**____?
- Text categorization
  - Spam filtering
  - Grammar guides
  - “Threat” analysis
  - Author/Plagiarism detection
- Recommendation engines (iTunes, Netflix, etc.)
- Face recognition
- Election/sports analysis (Moneyball, Nate Silver)
- Diagnostics, medical and otherwise

Classification

- y : variable for prediction (output)

- x : variable for observation (input)

- Training Data = Collection of $(x,\ y)$ pairs

- Machine Learning:
    - Given the training data, learn a mapping function $f(x){=}y$ that can map input variables to output variables

Features

- Input, x, is a vector of features
- Feature set is usually selected by a human
- Machine Learning algorithm:
  - How important is each feature to categorization?
  - Generate vector of these weights

Features (cont.)

- Some art to the selection of these
- Must be in a machine-readable form
- Often iterate through different choices
- Example features for text classification:
  - average word length
  - punctuation frequency
  - average sentence length
  - sentence structure
  - …

Supervised-Learning

- Define some “training” data (which contains the, usually human-defined, “truth”)

- Develop the weighting vector and create classifications using this data

- Applying the weighting and classifications to unknown data

- Goal is often *prediction* of unknown/unmeasured features

Unsupervised Learning

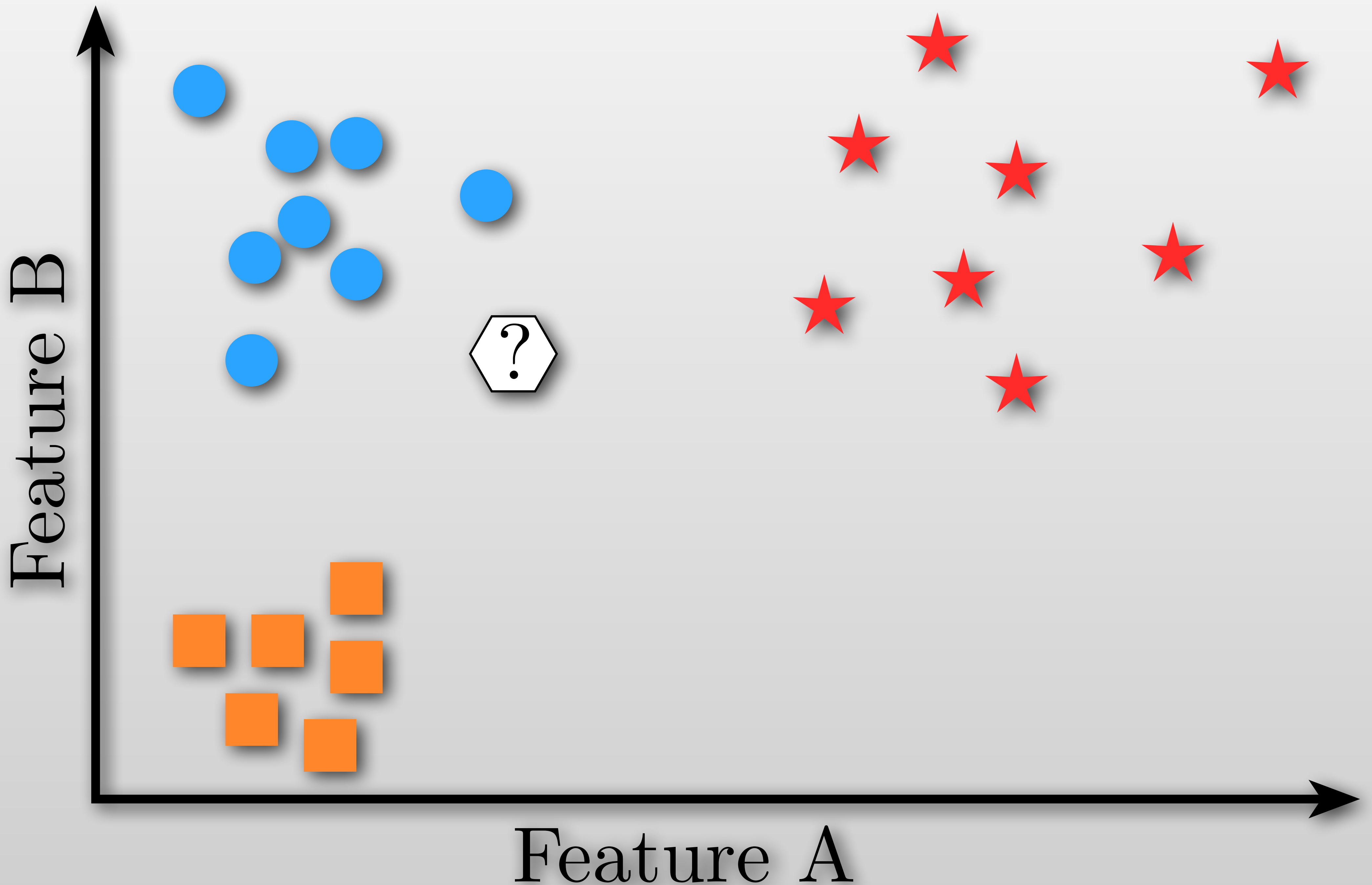
- Need to learn from data alone (no training)
- No human interaction (so cheaper, humans are expensive)
- Goal is most often categorization of existing data

Common Problems

- Overfitting:
  - Learning too specific to “training” data
  - Doesn’t allow generalization
  - Can result from improper feature selection
- Feature dominance
  - Weighting factors chosen such that one feature dominates
  - Most algorithms have methods to prevent one feature from dominating

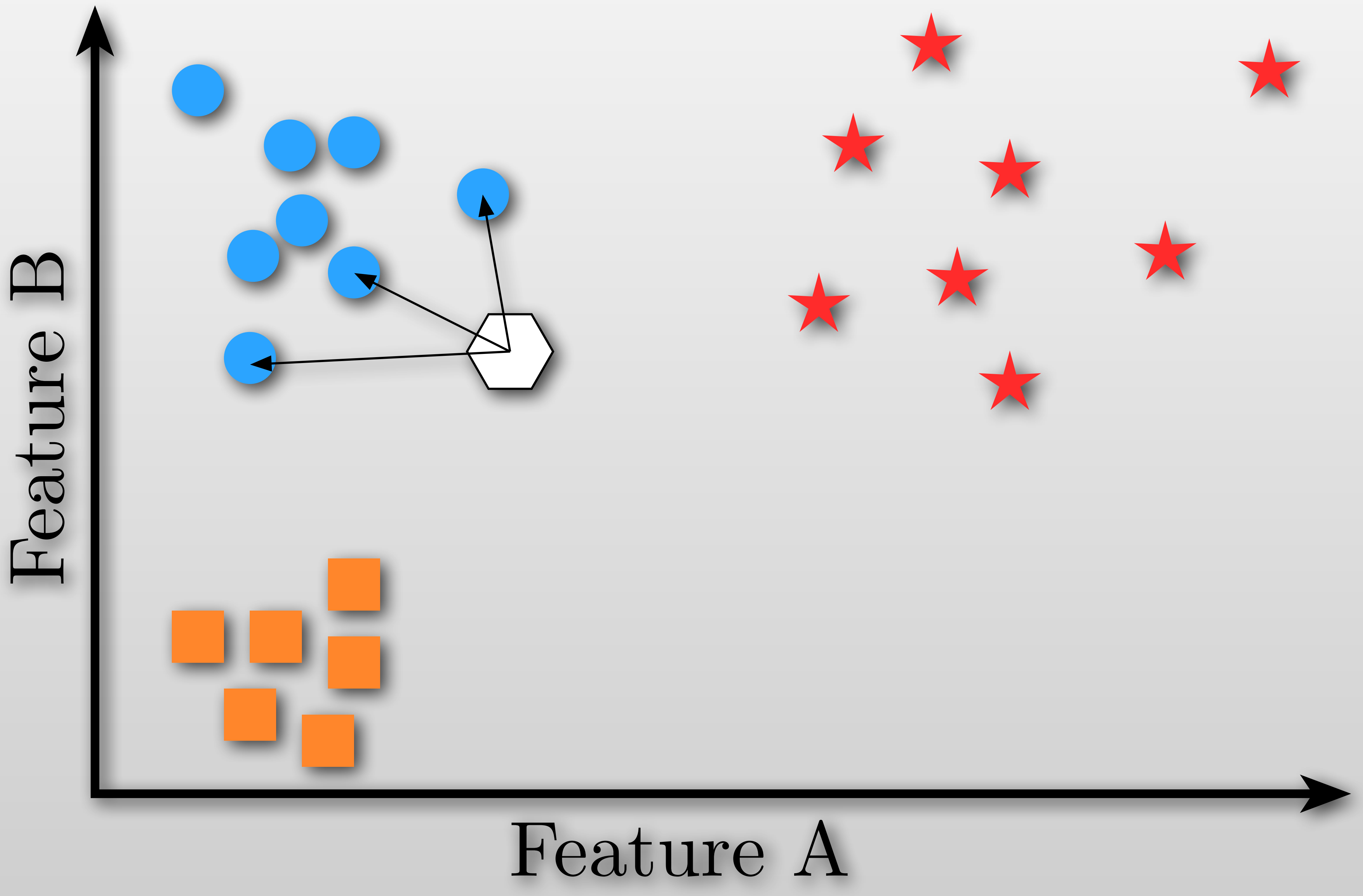
k-Nearest Neighbor

- “Distance”-based unsupervised classification



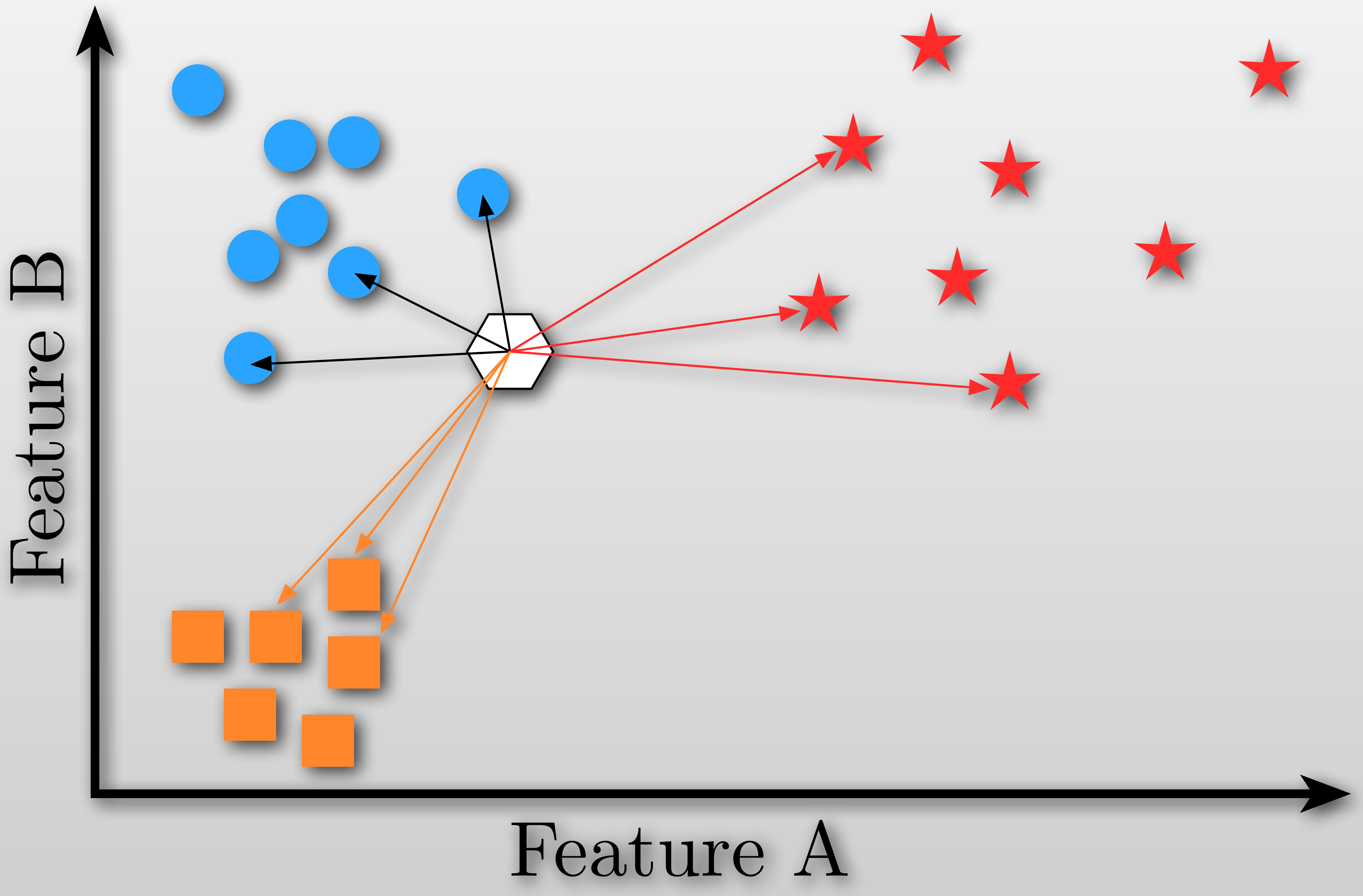
k-Nearest Neighbor (k=3)

- Compute the "distance" to k-nearest of each set



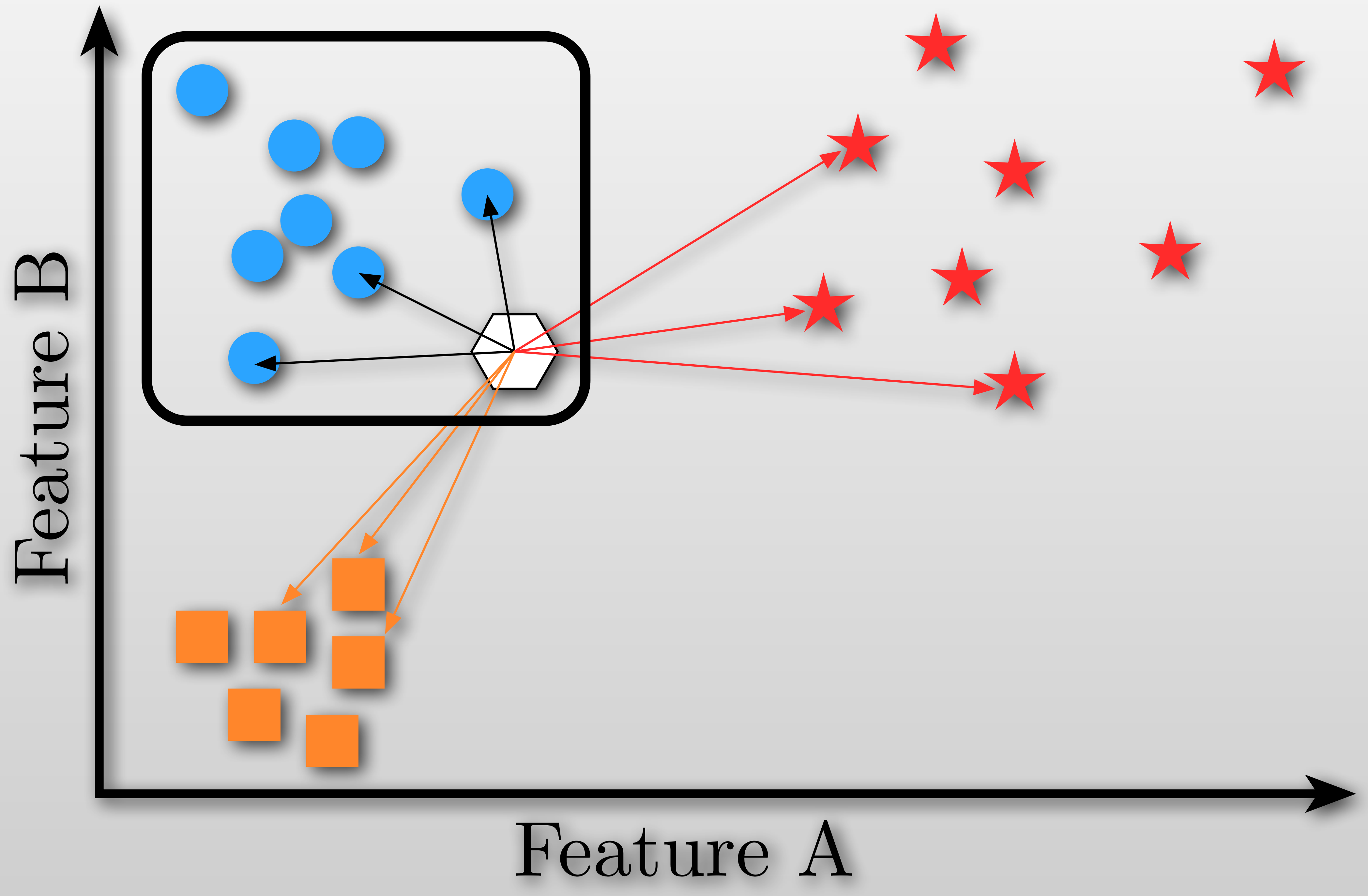
k-Nearest Neighbor (k=3)

- Categorize in “nearest” set



k-Nearest Neighbor (k=3)

- Categorize in “nearest” set



Problems?

- Can become computationally “expensive” with large data sets
- One dimension can dominate (normalize)

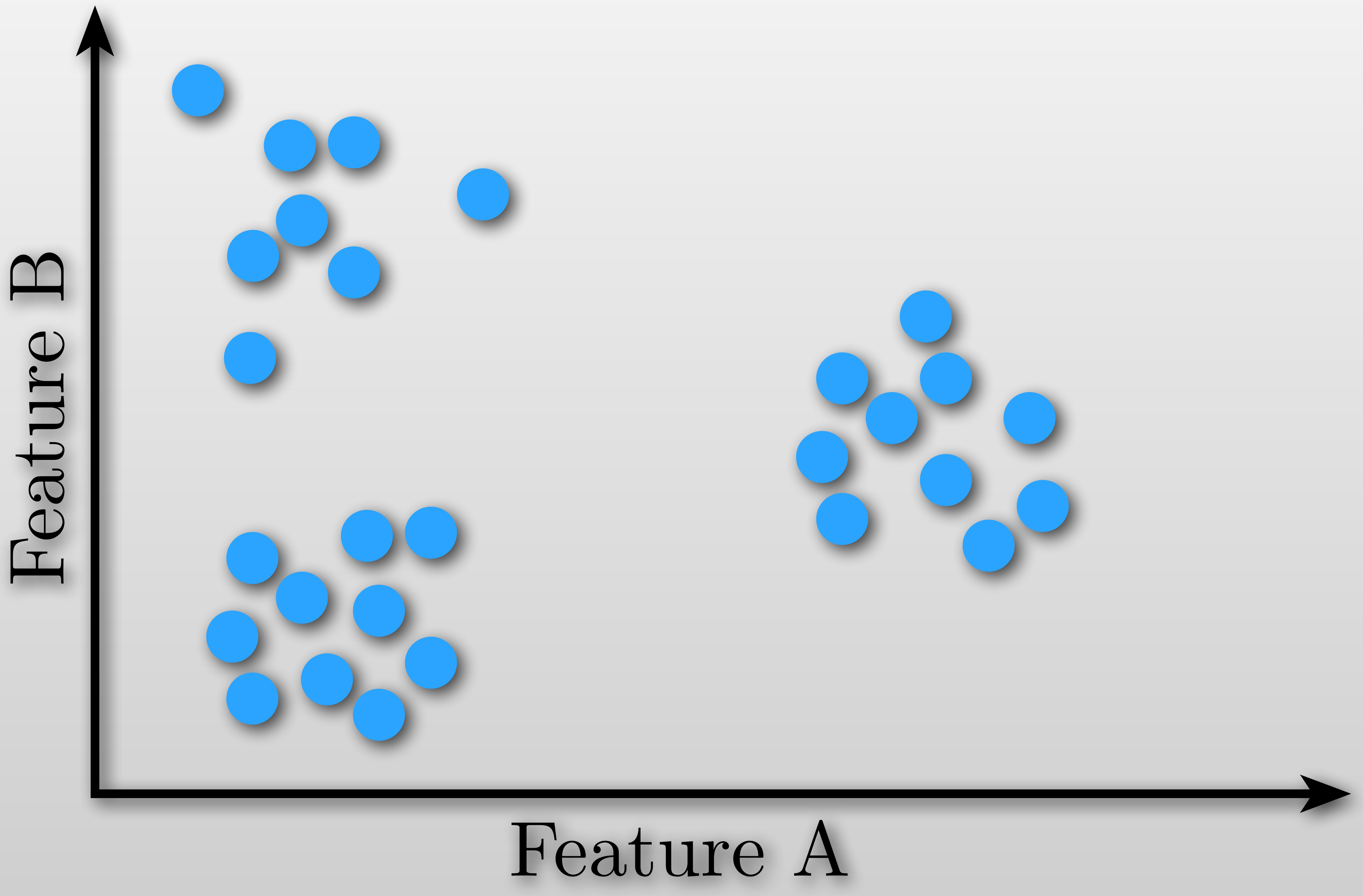
Improvements?

- Iterate over different values of k
  - Improve classification
  - *Check* of classification

- Use differences in distance as “quality” of fit

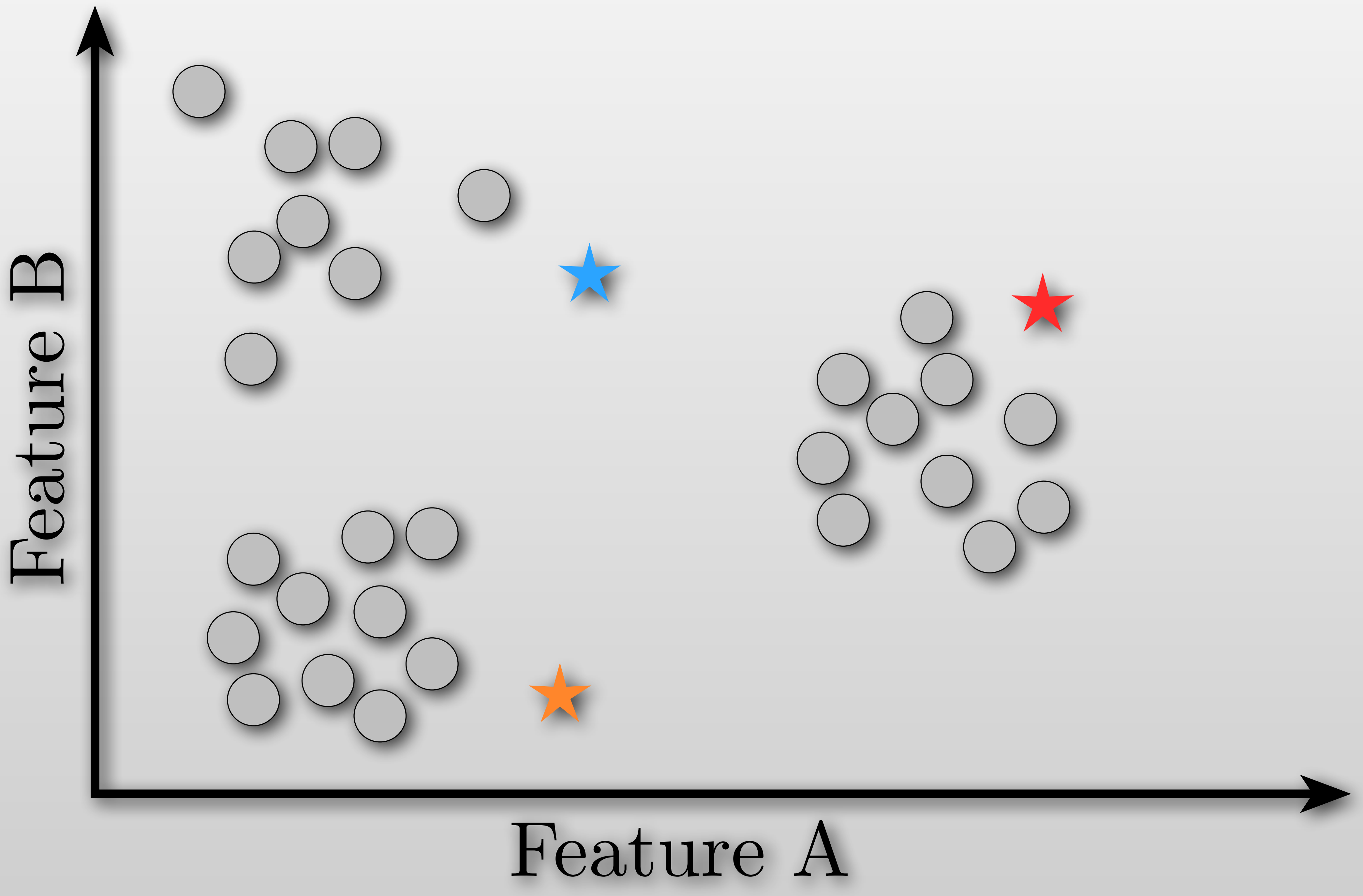
k-Means Clustering

- Classification not known *a priori*



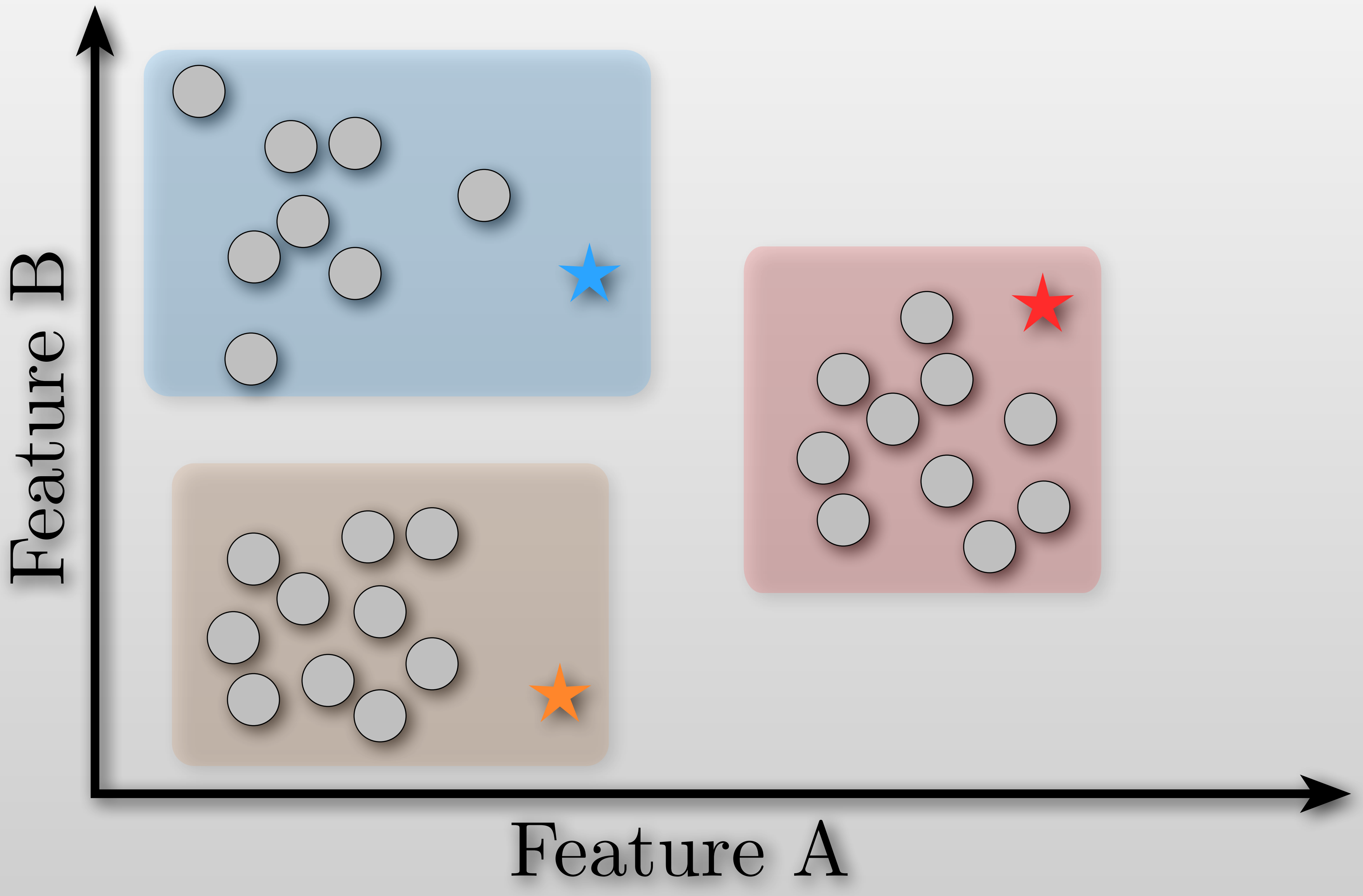
k-Means Clustering

- Pick k "centers"



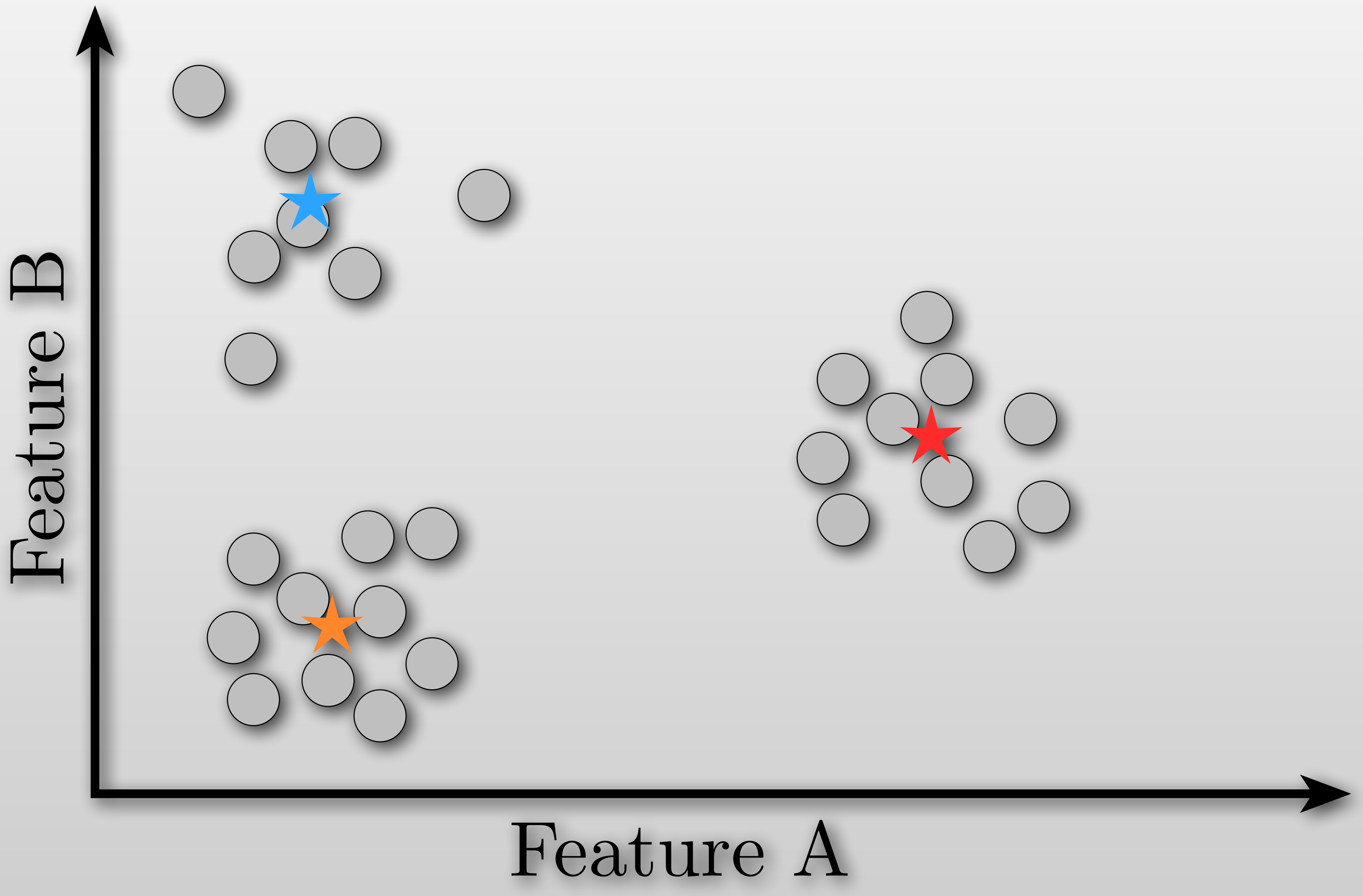
k-Means Clustering

- Associate each point with nearest center



k-Means Clustering

- New center is center of those associated points

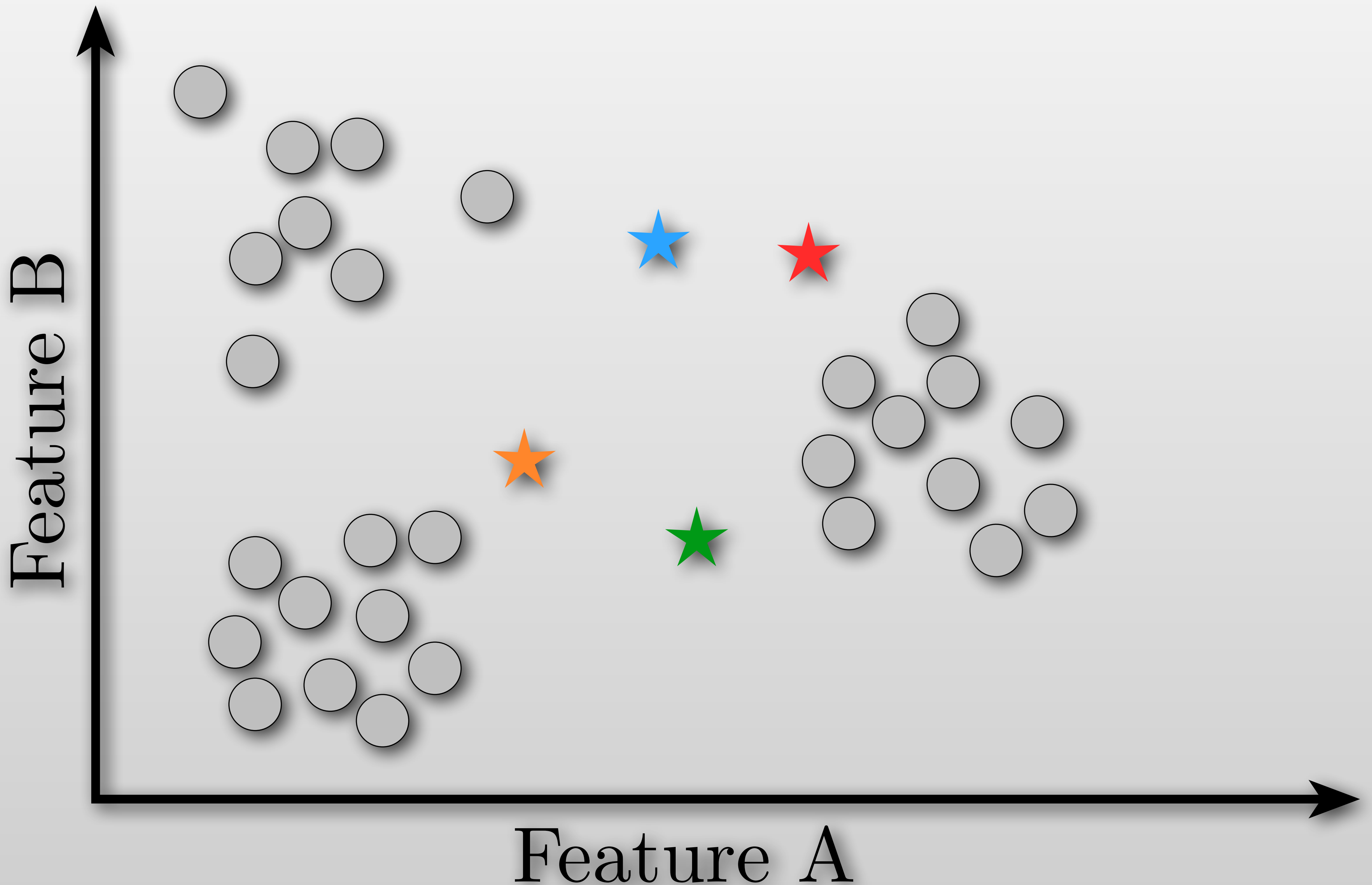


Problems

- How many k (categories)?
- Data “clusters” need to be similar size
- Needs iterative processing (can’t run just once)
- No guarantee of convergence

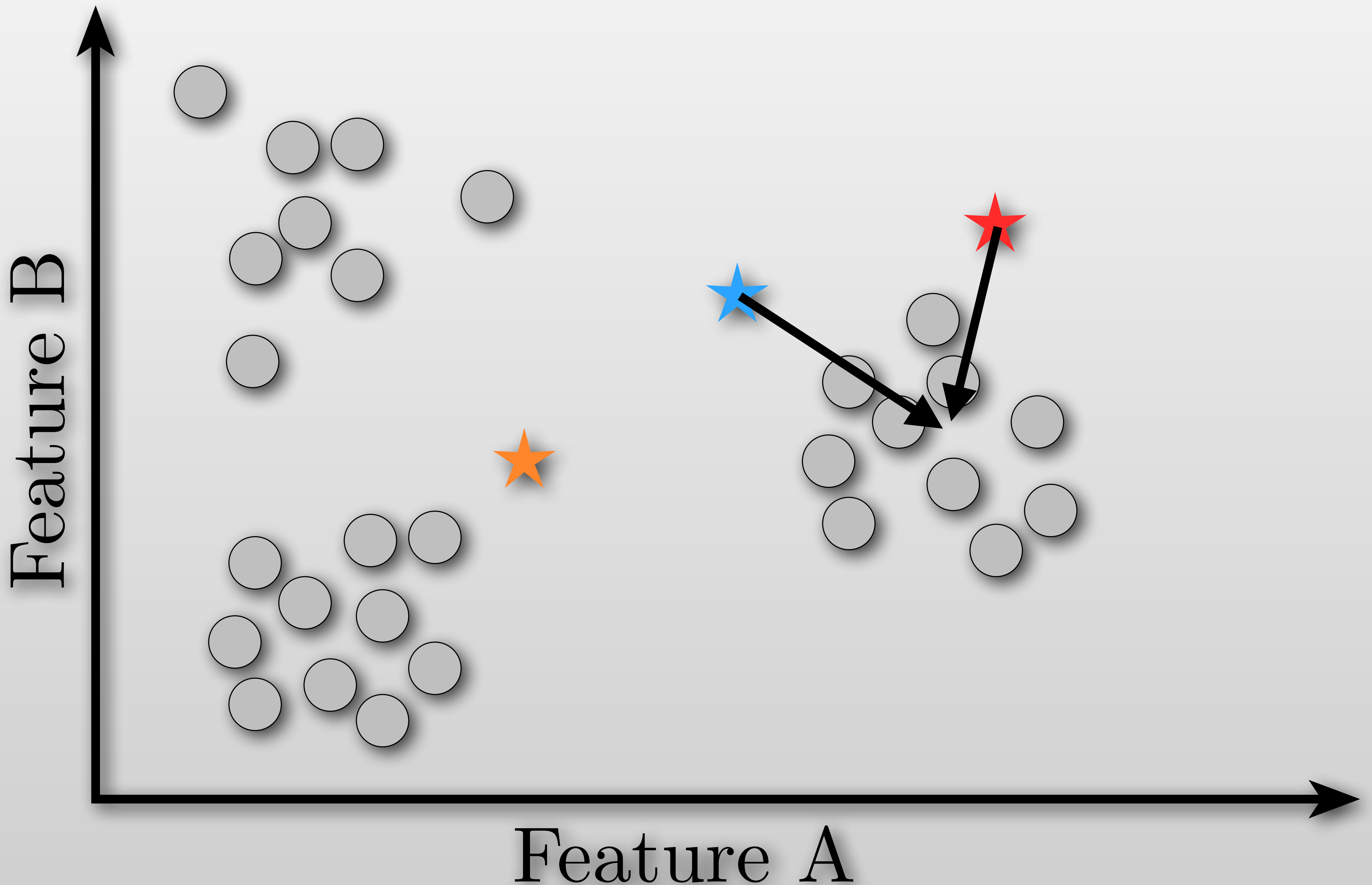
k-Means, “wrong” k

- Human supervisor can catch



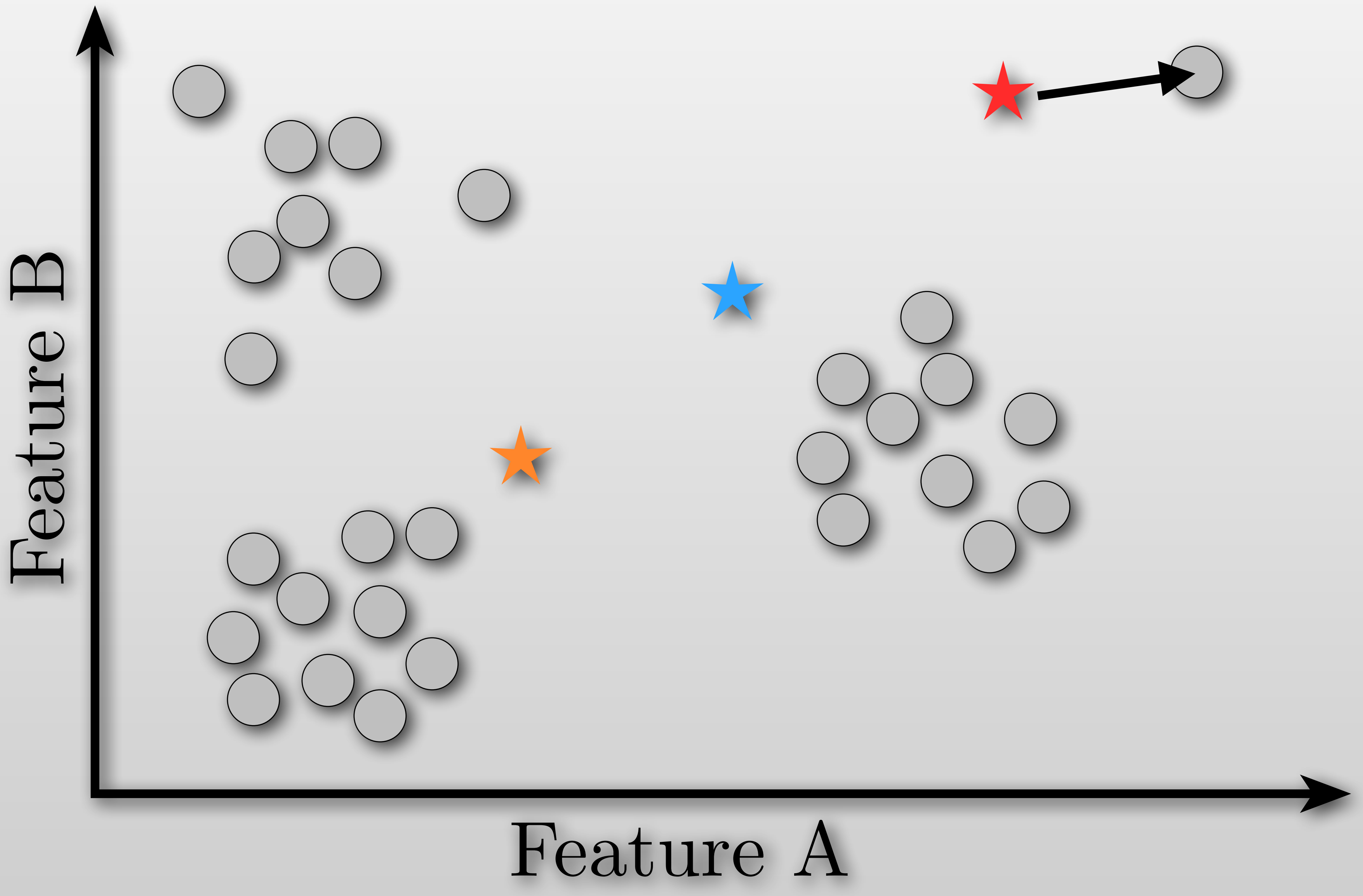
k-Means Doubling

- Two initial centers can converge to same category



k-Means Outliers

- Center can converge to outlier data



Other Challenges

- How to categorize this?