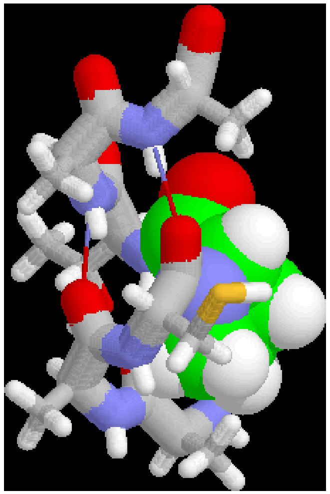
Chapter 2. Classification

Examples of Classification Task

- Predicting tumor cells as benign or malignant

- Classifying credit card transactions as legitimate or fraudulent



- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Recognize handwritten letters/digits

Work as waiter/waitress

Given many examples of boys/girls work as waiter/waitress

For a new boy, predict whether he will work as waiter

All examples/samples are represented by attribute vectors

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	*Yes*
X_2	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	*No*
X_3	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	*Yes*
X_4	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*10–30*	*Yes*
X_5	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	*No*
X_6	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	*Yes*
X_7	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	*No*
X_8	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	*Yes*
X_9	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	*No*
X_{10}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	*No*
X_{11}	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	*No*
X_{12}	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	*Yes*

People apply loan to buy house.

Given many examples of loan applications

For a new loan application, decide approve (or not) the loan

All examples/samples are represented by attribute vectors

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

IRS decide whether people cheat on income tax

Given many examples of past income tax returns/cheated or not

For a new people/tax return, decide cheated or not

All examples/samples are represented by attribute vectors

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Classification is similar to fitting data to a curve/function

Input: (x_1,y_1), (x_2,y_2), …, (x_n,y_n) The training data

Fit data to a curve/function: $y_i = f(x_i)$ Learning the model

For a new/query feature vector x, predict $y = f(x)$ classify x

Difference:

classification: y is often a class label (discrete, abstract)

data fitting : y is often real/integer value (height of a person)

Classification methods

- Use attribute/feature vectors
- Each data instance (document, image, sale-transaction, etc) is a point in the vector space

This is standard statistics framework
We can define "density", "distance", etc.

A classification method is

- model with parameters
- classifier
- function: y = f(x)
- learner (AI)
- learning system (machine learning)
- model parameters are learned from training data (teaching a learning model, parameter estimation)

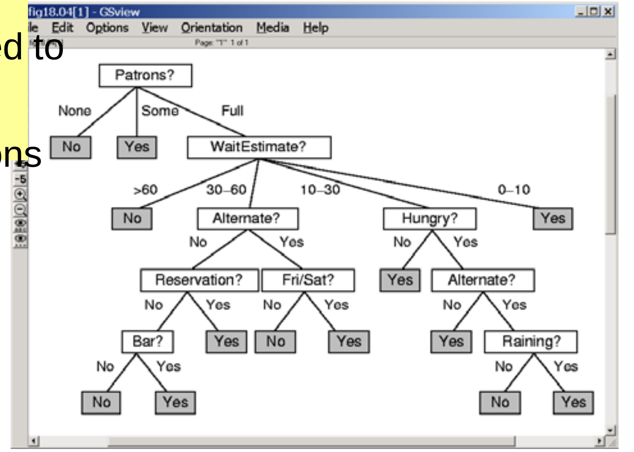
- Once a classifier is trained (model parameters are determined), it is used to assign class label to a new (query / test) document/image

- Clear distinction between training and testing

Uses different biases in predicting Russel's waiting habbits

- K-nearest neighbors

- Decision Trees
  - --Examples are used to
  - --Learn topology
  - --Order of questions



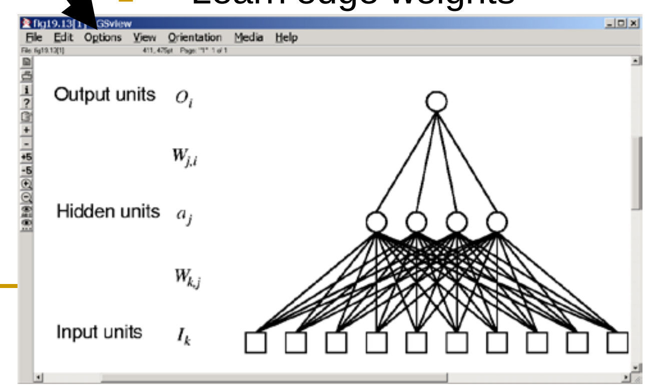
- If patrons=full and day=Friday
  - then wait (0.3/0.7)
- If wait>60 and Reservation=no
  - then wait (0.4/0.9)

- Association rules
- --Examples are used to
  - --Learn support and
  - confidence of association
  - rules

Example	Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	Yes
X_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	No
X_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	Yes
X_4	Yes	No	Yes	Yes	Full	$	No	No	Thai	10–30	Yes
X_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	No
X_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	Yes
X_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	No
X_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	Yes
X_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	No
X_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	No
X_{11}	No	No	No	No	None	$	No	No	Thai	0–10	No
X_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	Yes

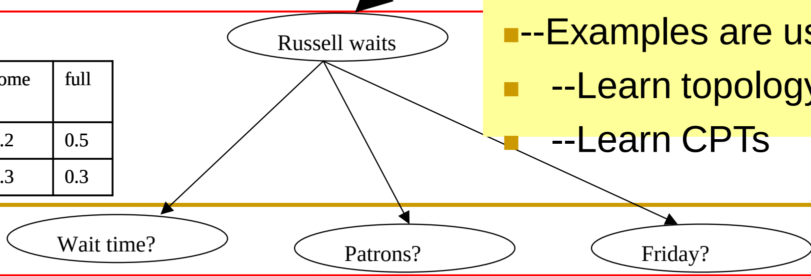
SVM

- Neural Nets
- --Examples are used to
  - --Learn topology
  - --Learn edge weights



- Naïve Bayes
- (bayesnet learning)
- --Examples are used to
  - --Learn topology
  - --Learn CPTs

RW	None	some	full
T	0.3	0.2	0.5
F	0.4	0.3	0.3



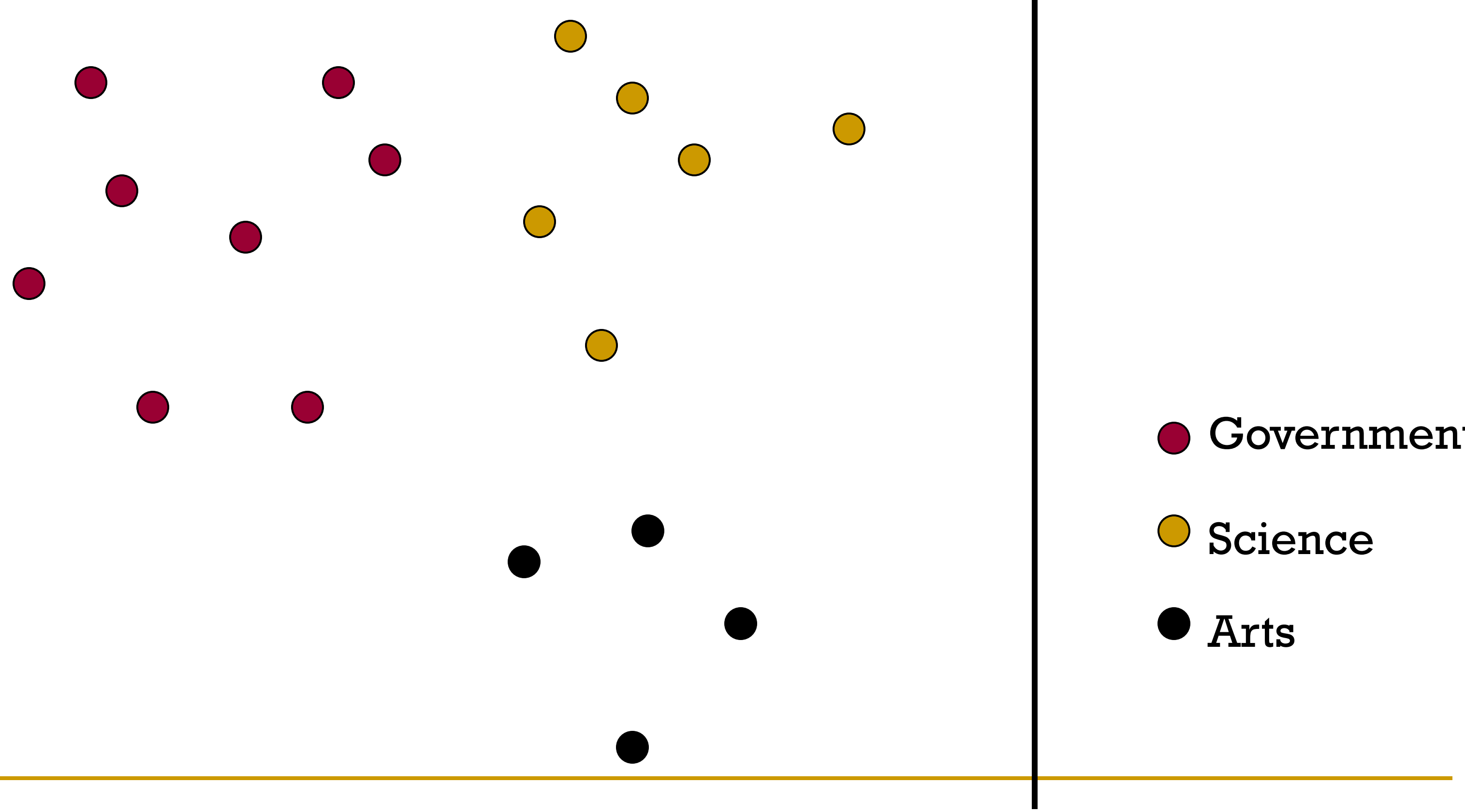
Chapter 2. Classification methods

- KNN
- Centroid Method
- Linear Regression
- Support Vector Machine

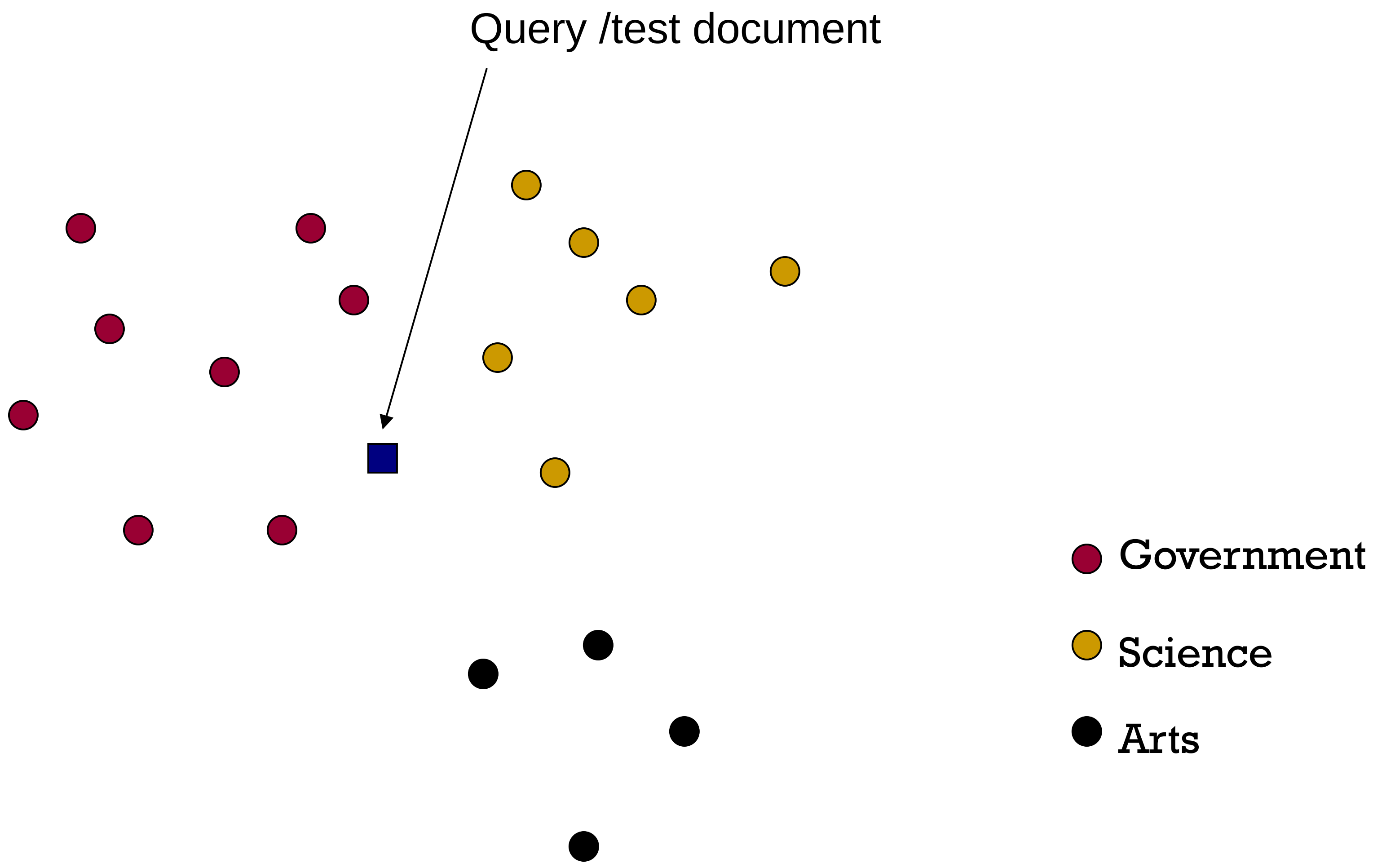
Classification is prediction

Example: Documents in a Vector Space

Existing data samples/examples



For a new/query/test document, which class it belongs to?



Classification methods

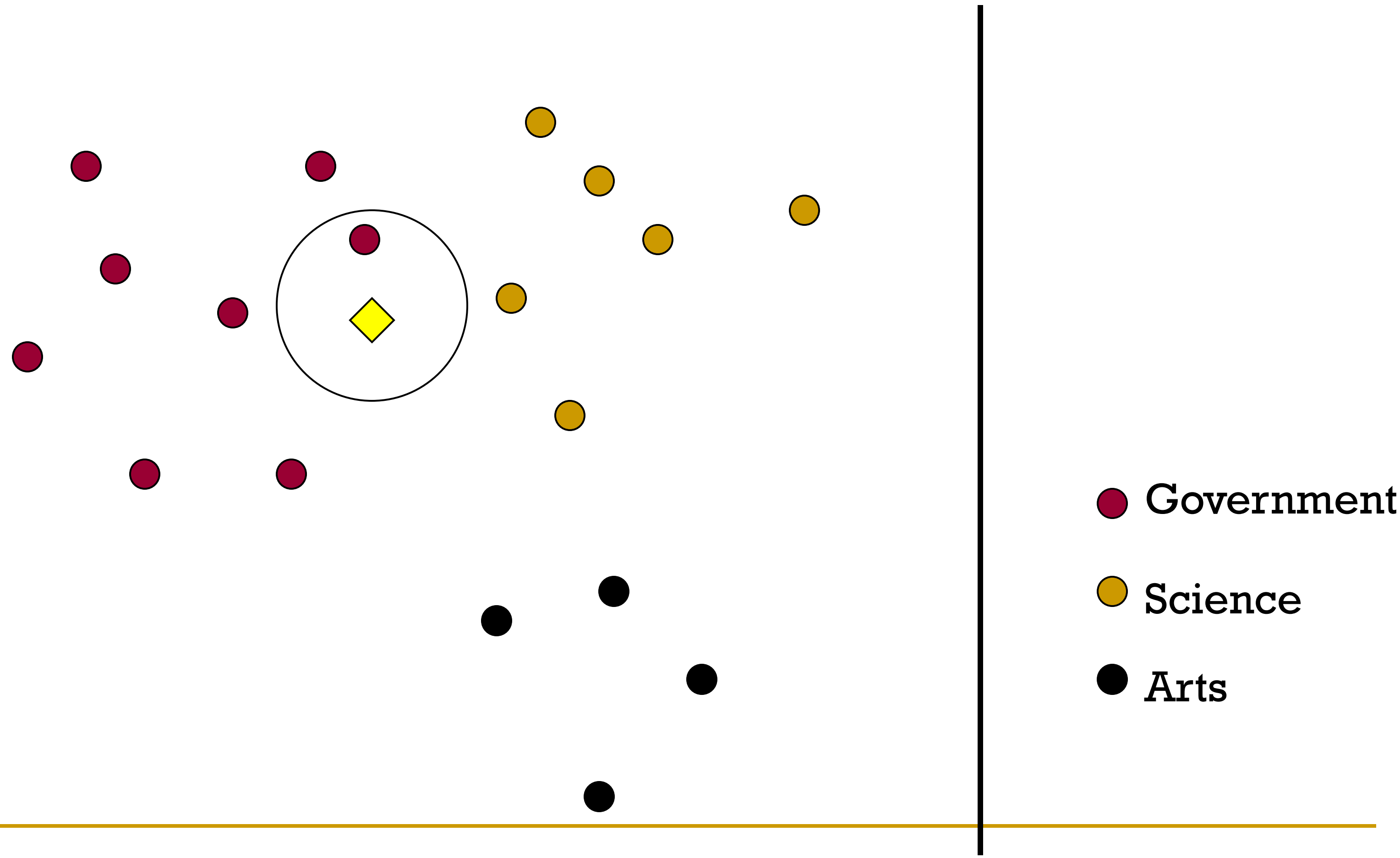
- **KNN**
- **Centroid Method**
- **Linear Regression**
- **Support Vector Machine**

k Nearest Neighbor Classification

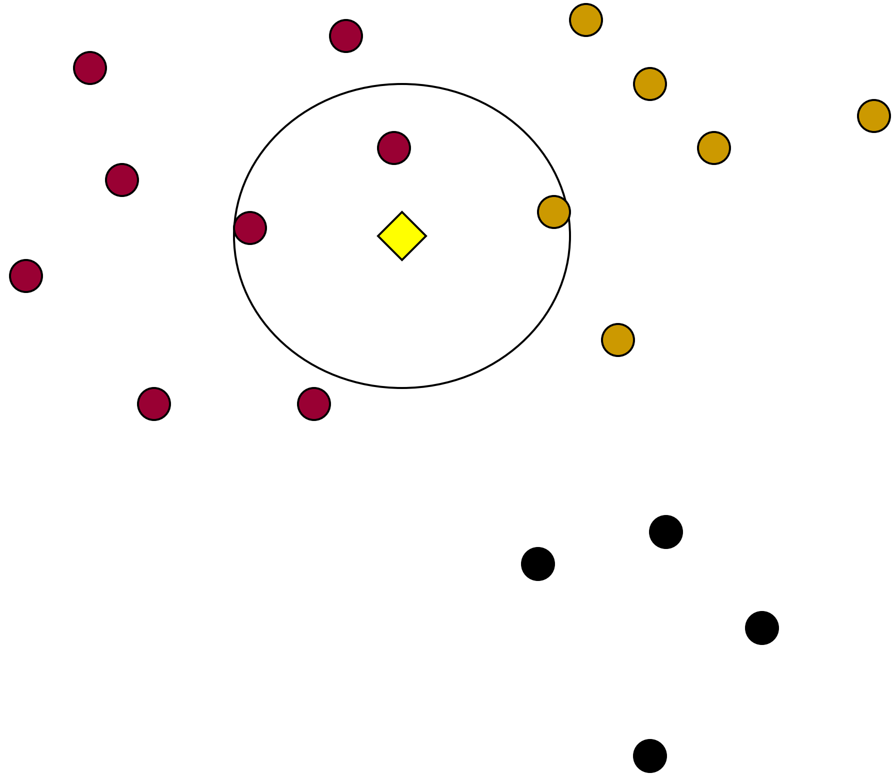
- kNN = k Nearest Neighbor

- To classify a data object *d* into a class c:
- Define *k*-neighborhood as *k* nearest neighbors of *d*
- Count number of data objects belonging to c [= q_c]
- Estimate Prob(c|*d*) = q_c / k
- Choose as class argmax_c P(c|*d*)   [= majority class]

Example: k=1 (1NN)



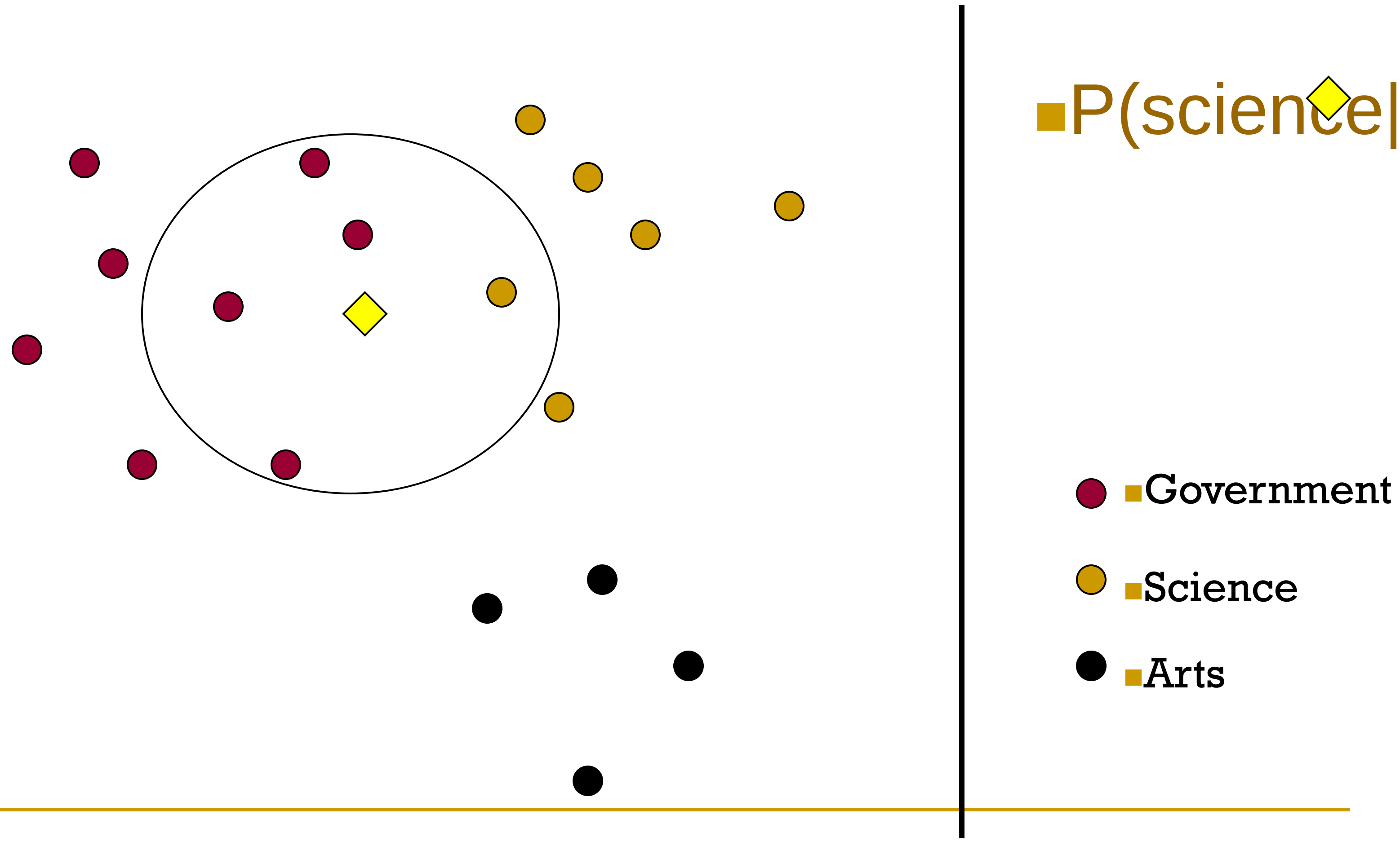
Example: k=3 (3NN)



- P(science|

- Government
- Science
- Arts

Example: k=5 (5NN)



KNN Learning Algorithm

- Rational: data points of same class distributed closeby
- Learning is to determine
  - k=?
  - distance/similarity metric to determine which one is closer
- Also called:
  - Case-based learning
  - Memory-based learning
  - Lazy learning

Distance / Similarity

- KNN use a distance metric.
- Euclidean distance for real-valued feature vectors.
- Hamming distance for category-valued feature vectors (=number of differring features)
- cosine similarity for document/query (vector space model)

k Nearest Neighbor (use larger k)

- Using only the closest examples to determine the class could have errors:
  - A single atypical (abnormal) example.
  - Noise (i.e., errors) in ground-truth class labels of a single training example.
- Solution: use larger k
- Value of *k* is typically odd; 3 and 5 are most common.

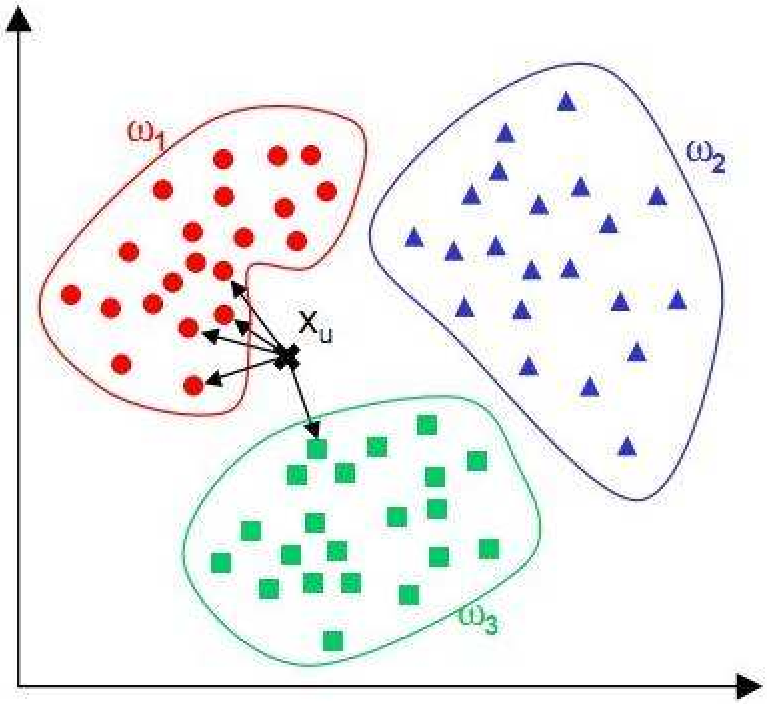
k Nearest Neighbor (**inefficient for big data**)

- Searching knn in database is linear (go thru all data)
- When database (existing data samples) are large, searching is inefficient

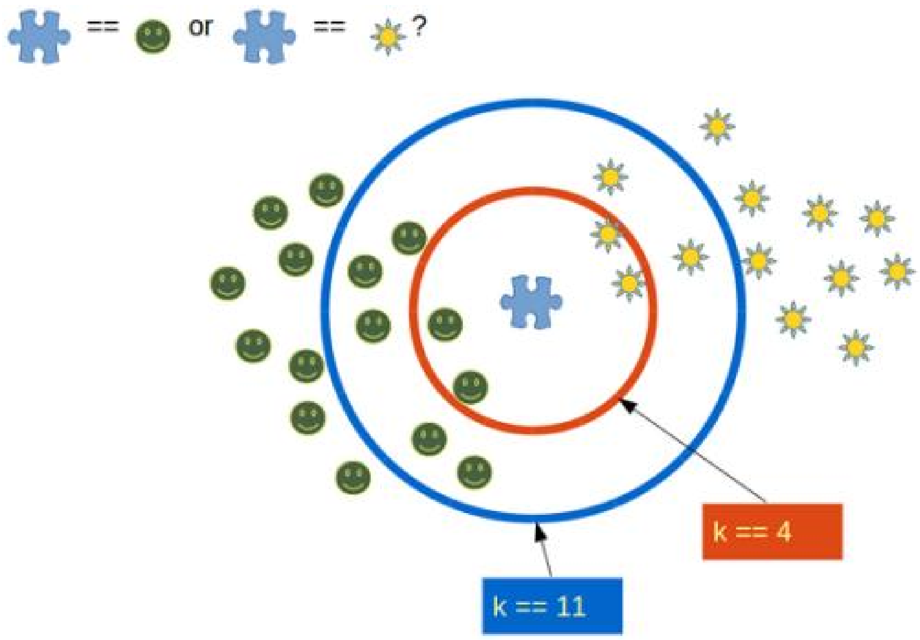
- Solution: divide data into groups; each groups is represented by an anchor. Search algorithm:
  - Finding the closest anchor
  - In the group represented by this anchor, find closest sample
  - This method is not exact, could have errors

kNN is Close to Optimal

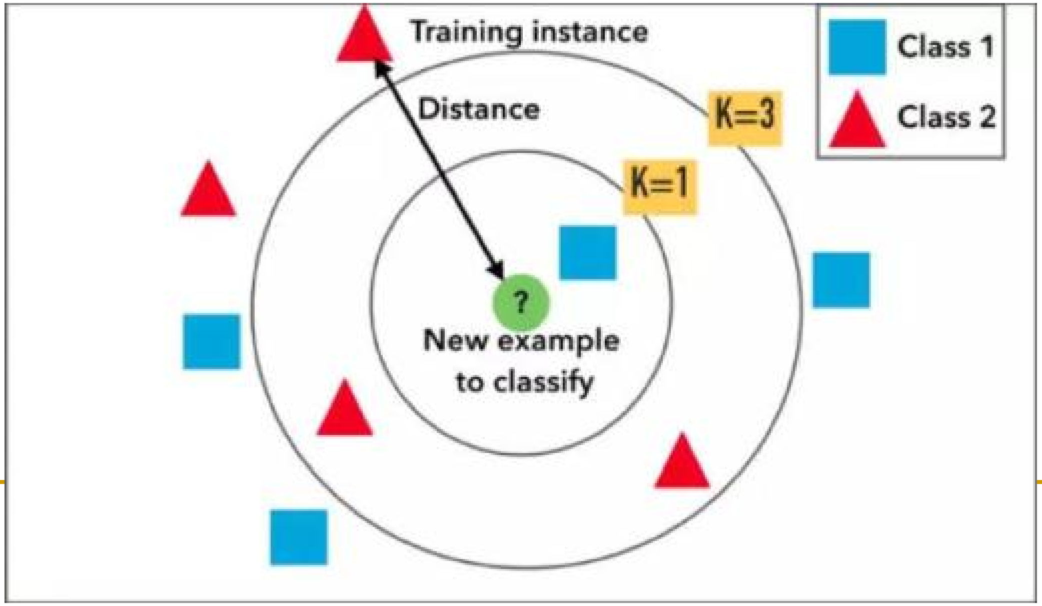
- E. Fix and J. L. Hodges, Jr., “Discriminatory analysis, non- parametric discrimination” 1951
- Cover and Hart (1967)
- Asymptotically, error rate of 1nn classification is less than twice the Bayes rate



ω_1
ω_2
ω_3
X_u



== or == ?
k == 4
k == 11



Training instance
Distance
K=3
K=1
Class 1
Class 2
?
New example
to classify

KNN in Recommender Systems

Recommender Systems

- **We live in a complex society: too many choices for everything:**
- **Need to Recommend**
  - **Books**
  - **Movies**
  - **Restaurants**
  - **Medicine**
  - **doctors**
  - **Vacations**
  - **…**

- **Need to build information systems for these tasks**
- **2D Recommendation Systems**
  - **Users, items (books, movies)**

Recommender Systems

- **Collaborative Filtering**
  - **Collecting large amount of data, user tastes. Recommend based similar tastes of similar users.**

CF: K Nearest Neighbor

	Hoop Dreams	Star Wars	Pretty Woman	Titanic	Blimp	Rocky XV
Joe	D	A	B	D	?	?
John	A	F	D		F	
Susan	A	A	A	A	A	A
Pat	D	A		C		
Jean	A	C	A	C		A
Ben	F	A				F
Nathan	D		A		A	

Collaborative Filtering, Herlocker, Konstan, Borchers, Riedl, SIGIR1999

CF: K Nearest Neighbor

	Hoop Dreams	Star Wars	Pretty Woman	Titanic	Blimp	Rocky XV
Joe	D	A	B	D	?	?
John	A	F	D		F	
Susan	A	A	A	A	A	A
Pat	D	A		C		
Jean	A	C	A	C		A
Ben	F	A				F
Nathan	D		A		A	

CF: K Nearest Neighbor

	Hoop Dreams	Star Wars	Pretty Woman	Titanic	Blimp	Rocky XV
Joe	D	A	B	D	?	?
John	A	F	D		F	
Susan	A	A	A	A	A	A
Pat	D	A		C		
Jean	A	C	A	C		A
Ben	F	A				F
Nathan	D		A		A	

Movie Rating Recommender System (2D)

INPUT: A user gives ratings (1-5) to several movies
. (typically 5 to 20)

OUTPUT: Based on this limited information, the system
. provides ratings of all movies

Mathematically,

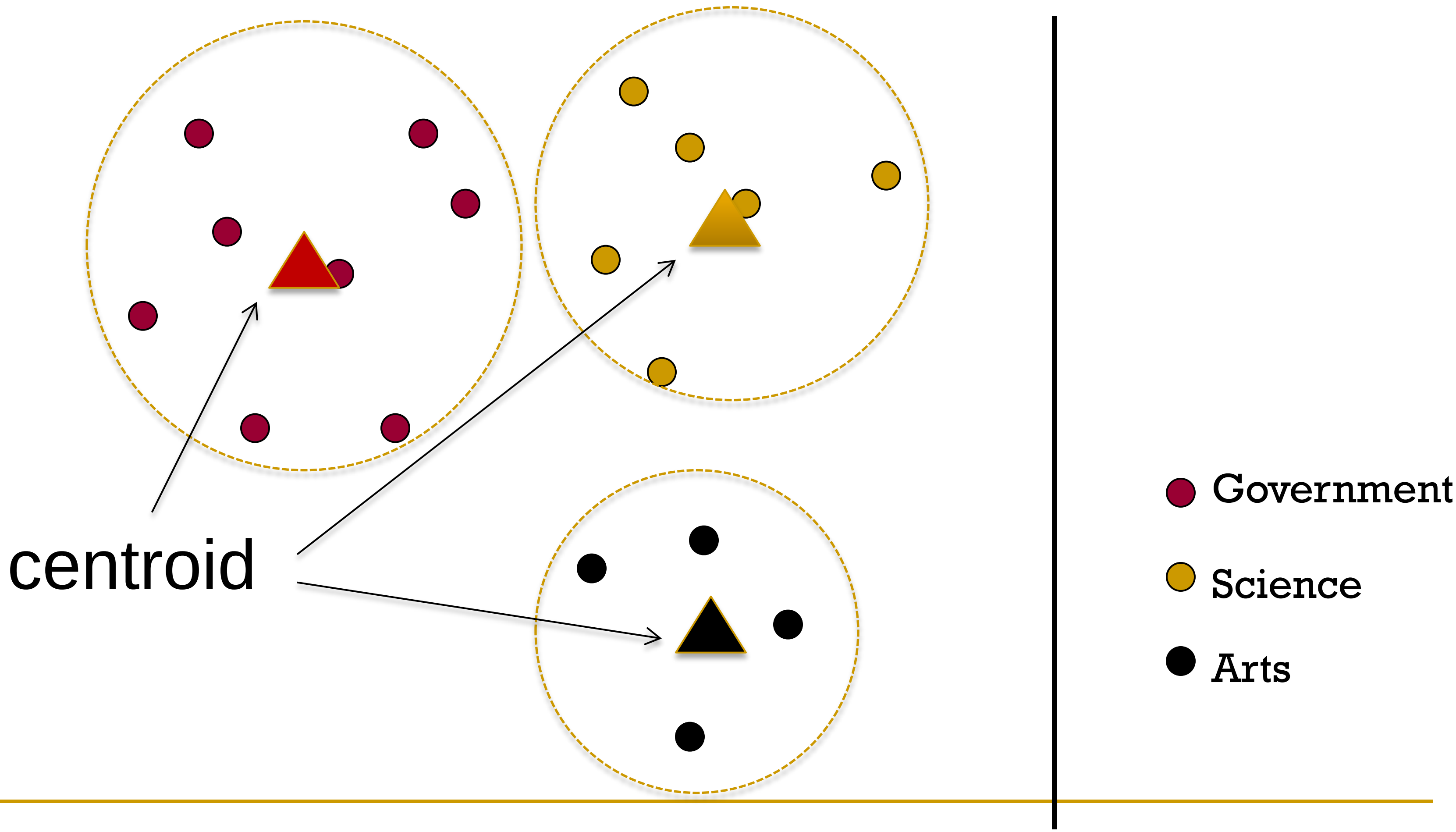
input =(1, ?, ?, ?, 5, 2, ?, ?, ? ... ?, 2, ?, ? ...)
Output=(1, 3, 1, 2, 5, 2, 3, 2, 1 ... 5, 2, 4, 2 ...)

Classification methods

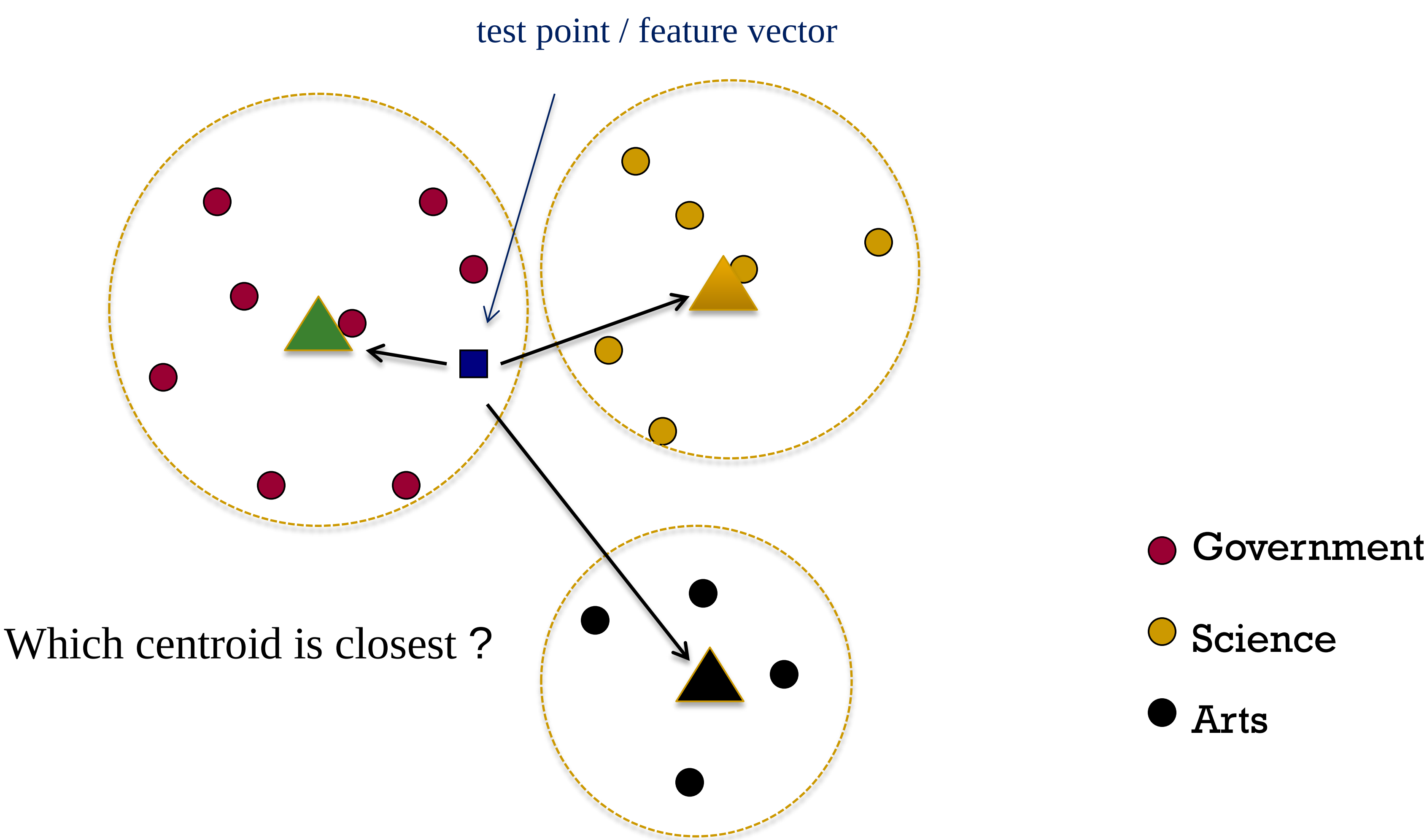
- KNN
- **Centroid Method**
- Linear Regression
- Support Vector Machine

Centroid Classification Method

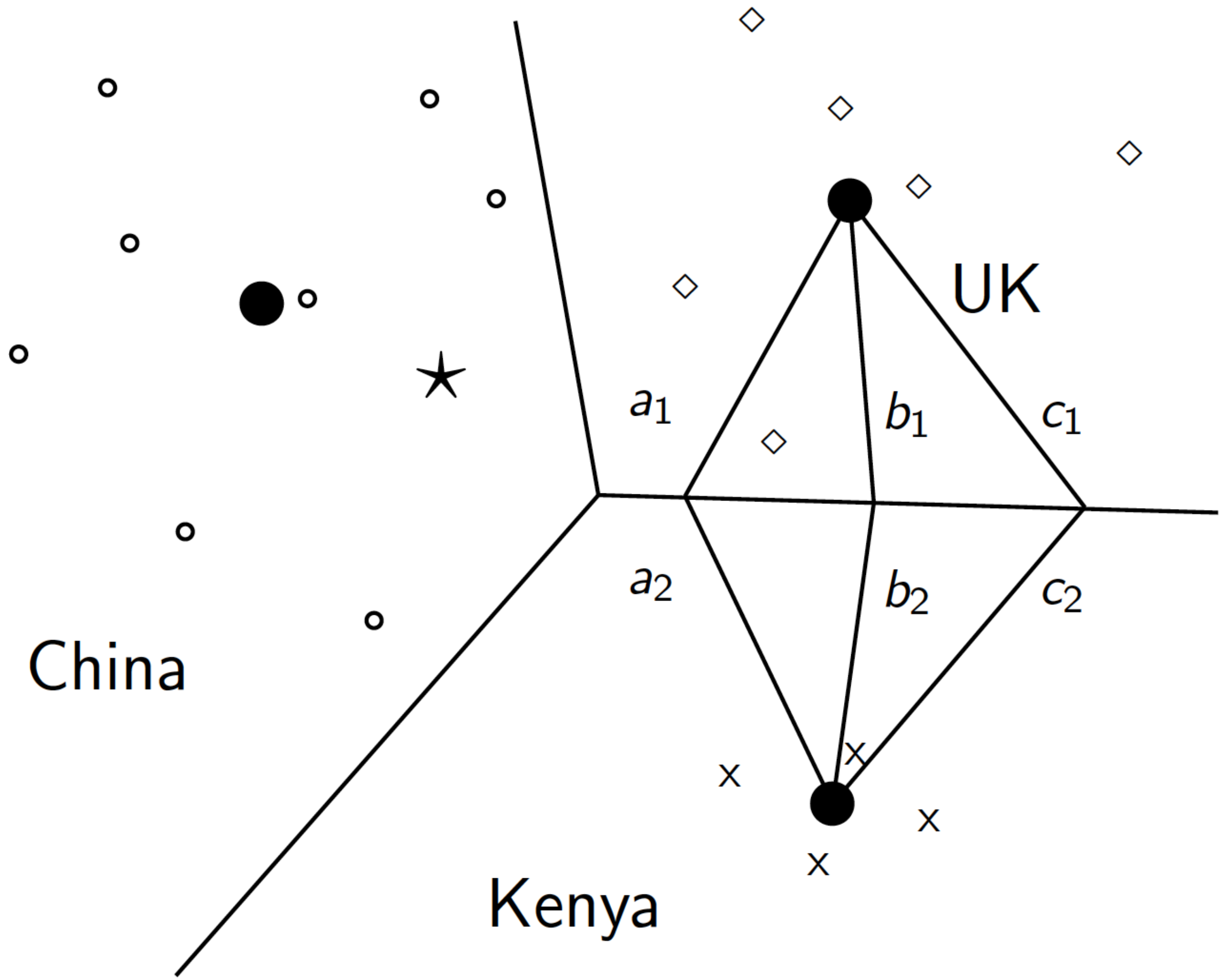
(The centroid of class k is the class average over all feature vectors in class k.
A centroid is also feature vector, but often not an original data objects)



Centroid classification method



Centroid method in 3D



Centroid method

- Test point is compared to k centroids
- Faster than kNN
  - because number-of-class < number-of-data-vectors
- Typically less accurate than kNN
- Historically is called Rochioo algorithm in 1970s, but re-invented many times later

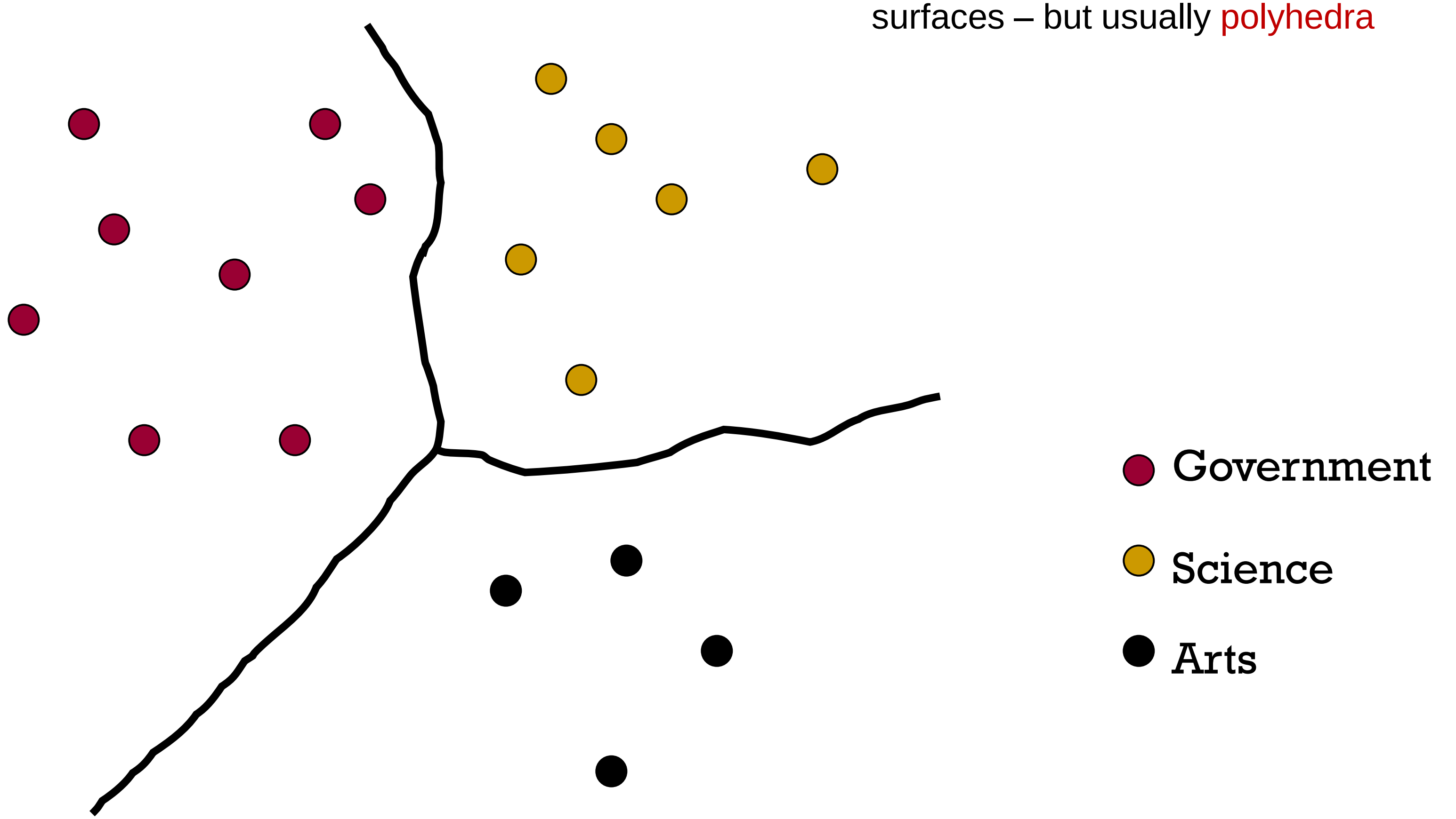
A classification method is a function
A function defines class boundary

Class boundary also is called decision boundary/surface

Decision boundary is a broadly used concept

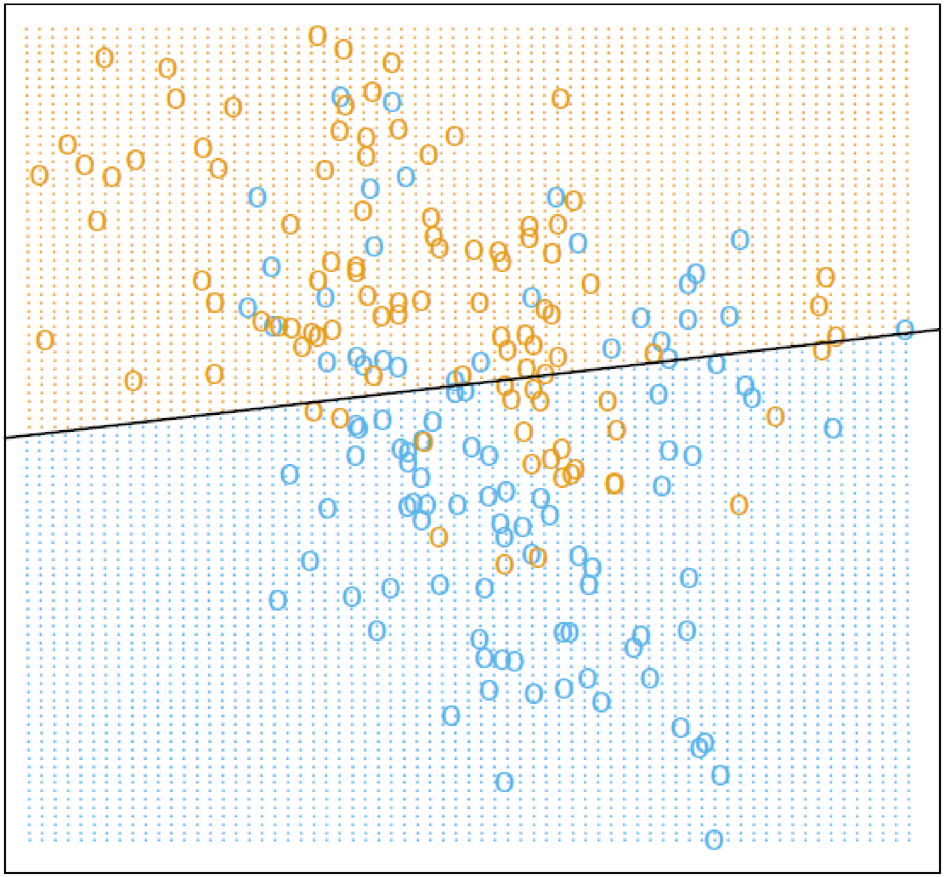
kNN decision boundaries

Boundaries are in principle arbitrary surfaces – but usually polyhedra

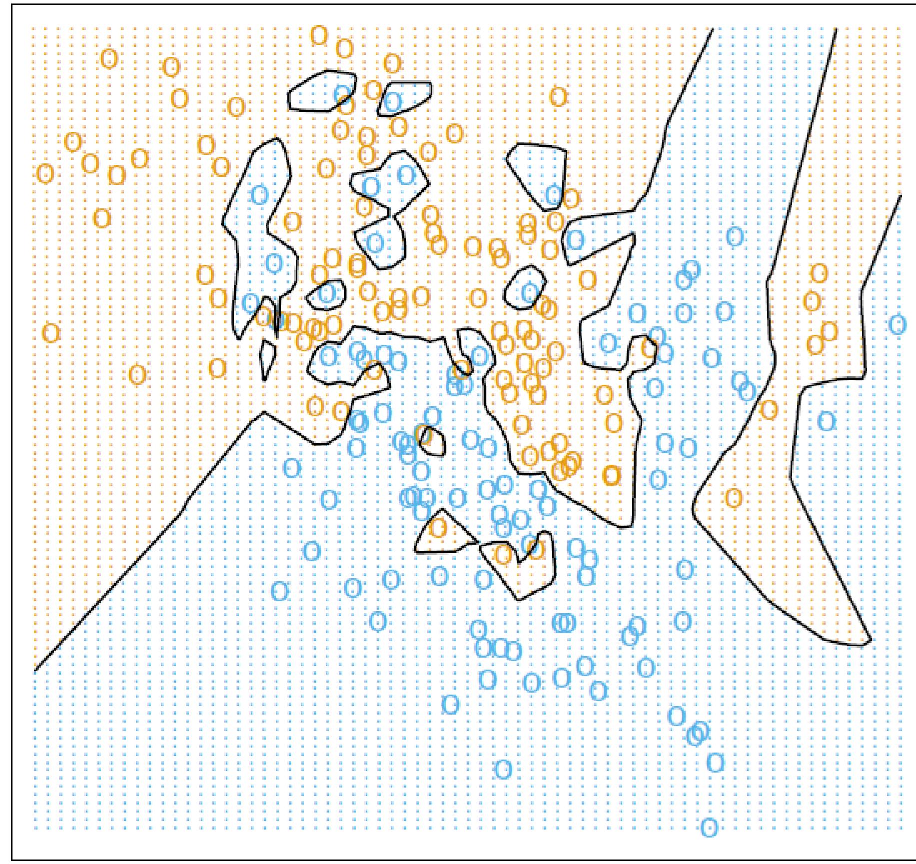


kNN gives locally defined decision boundaries between classes – far away points do not influence classification decision

Decision Boundary

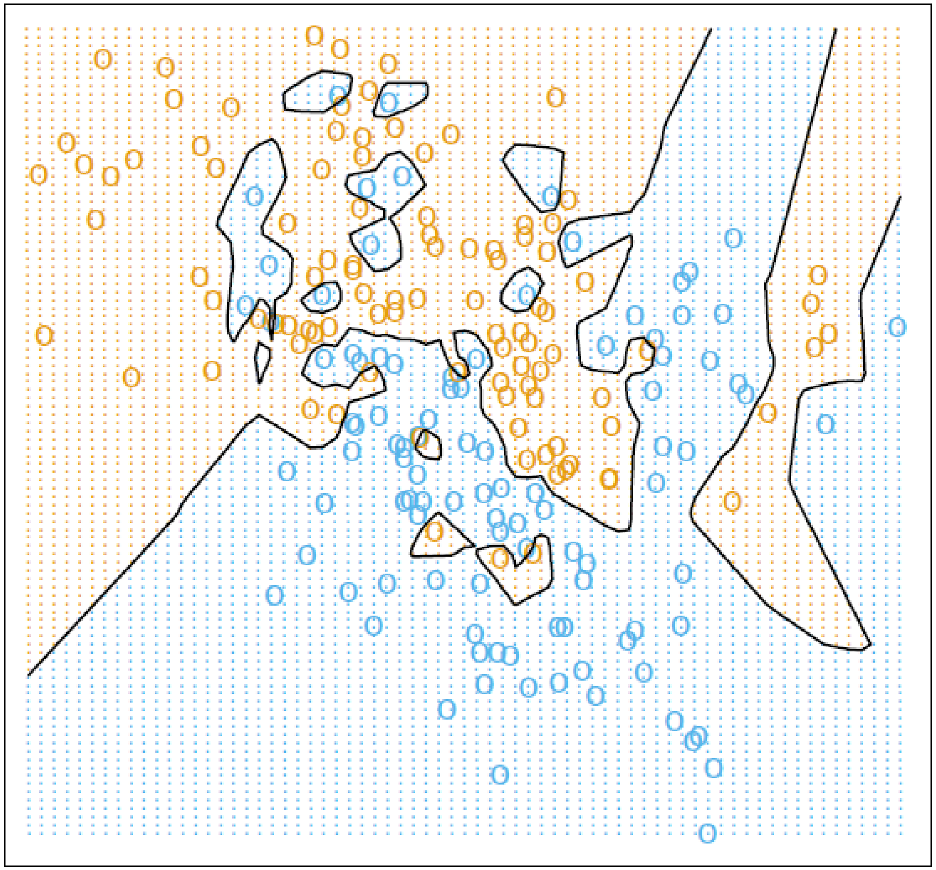


Linear Regression

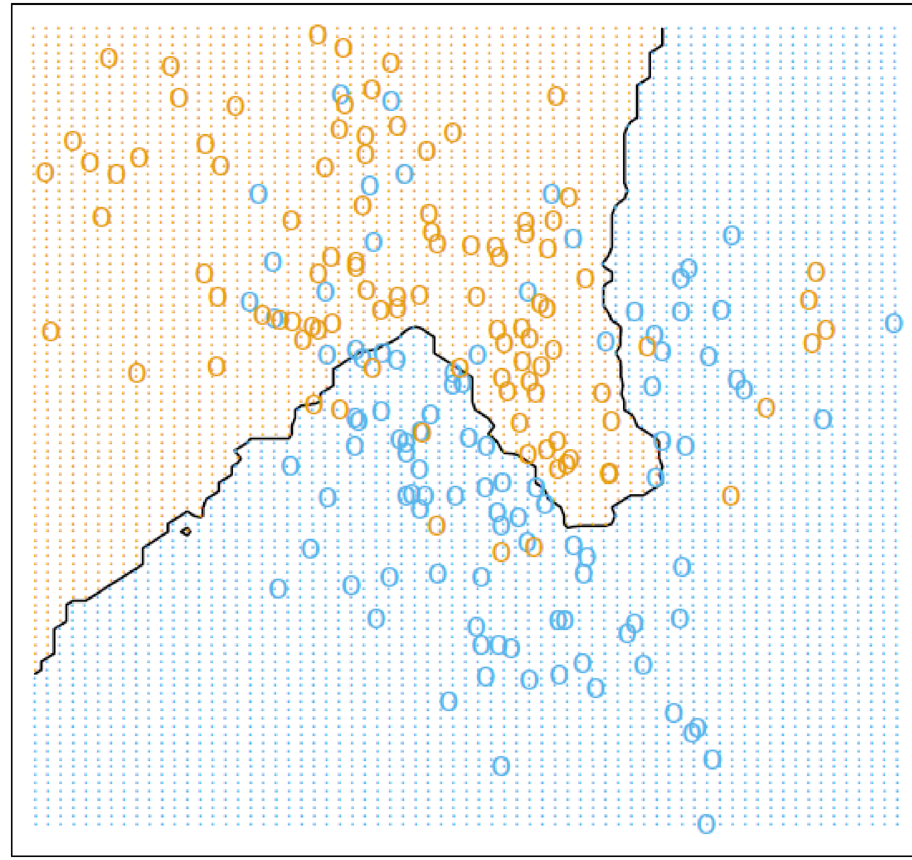


1nn

Decision Boundary

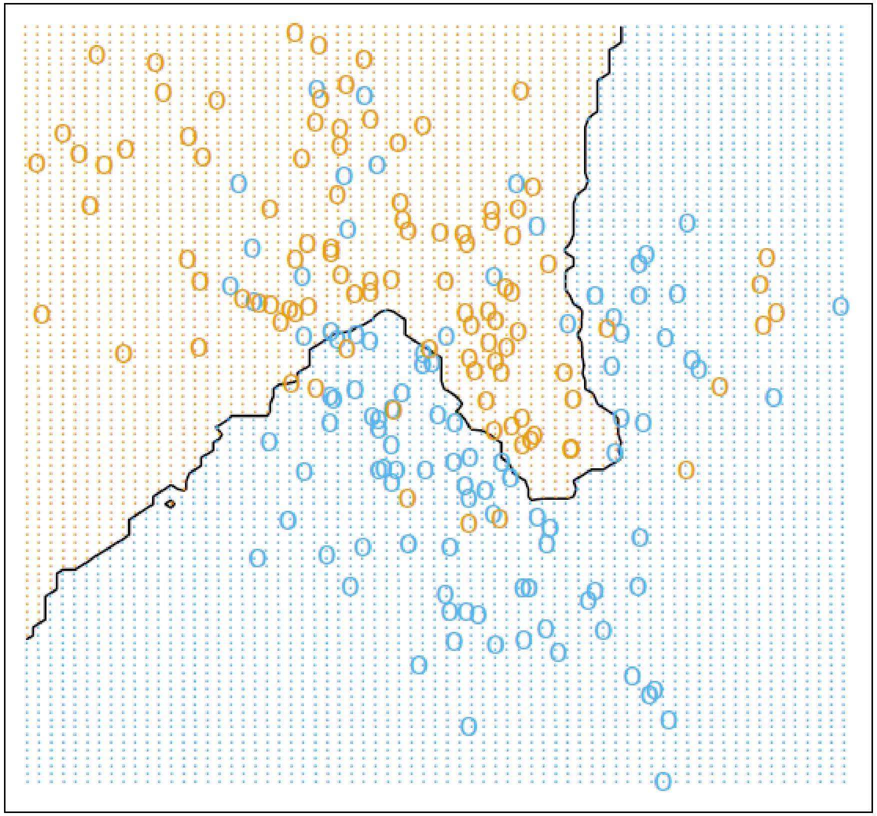


1nn

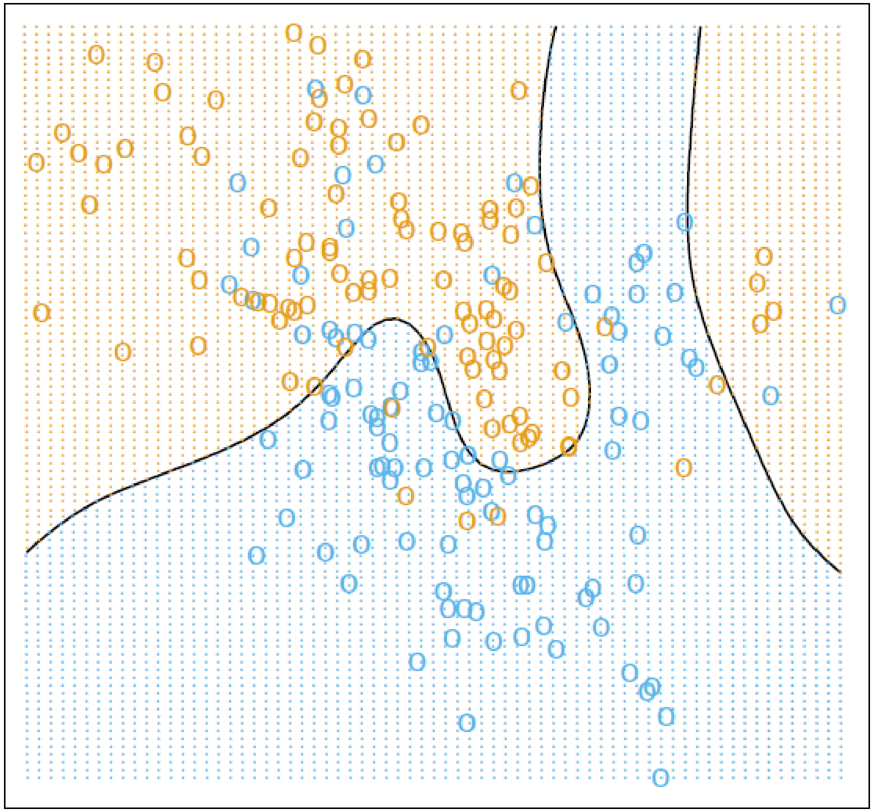


15nn

Decision Boundary



15nn



Bayes Optimal boundary

Linear Classification methods

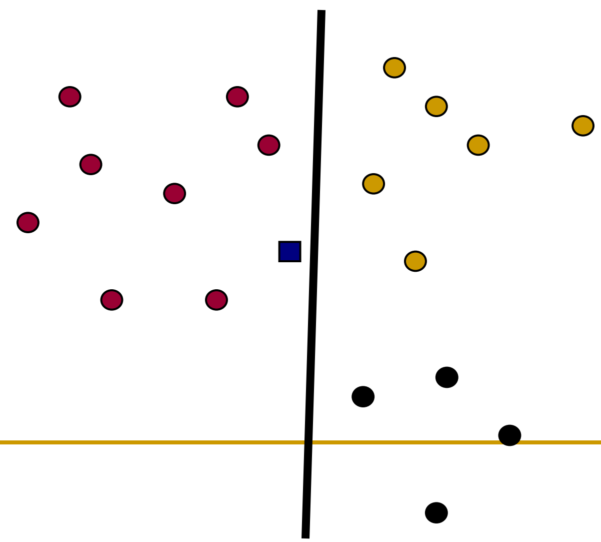
- A linear classification uses a linear function
  - Therefore, class boundaries are hyperplanes
  - Is KNN a linear classifier? No
  - Is Centroid method a linear classifier? No

- 2-class classification problems most naturally uses linear classification
  - A line (plane/hyperplane) separates two classes
  - (divide the feature-space into two parts)
- Linear classification methods
  - Linear Regression
  - Support vector machine

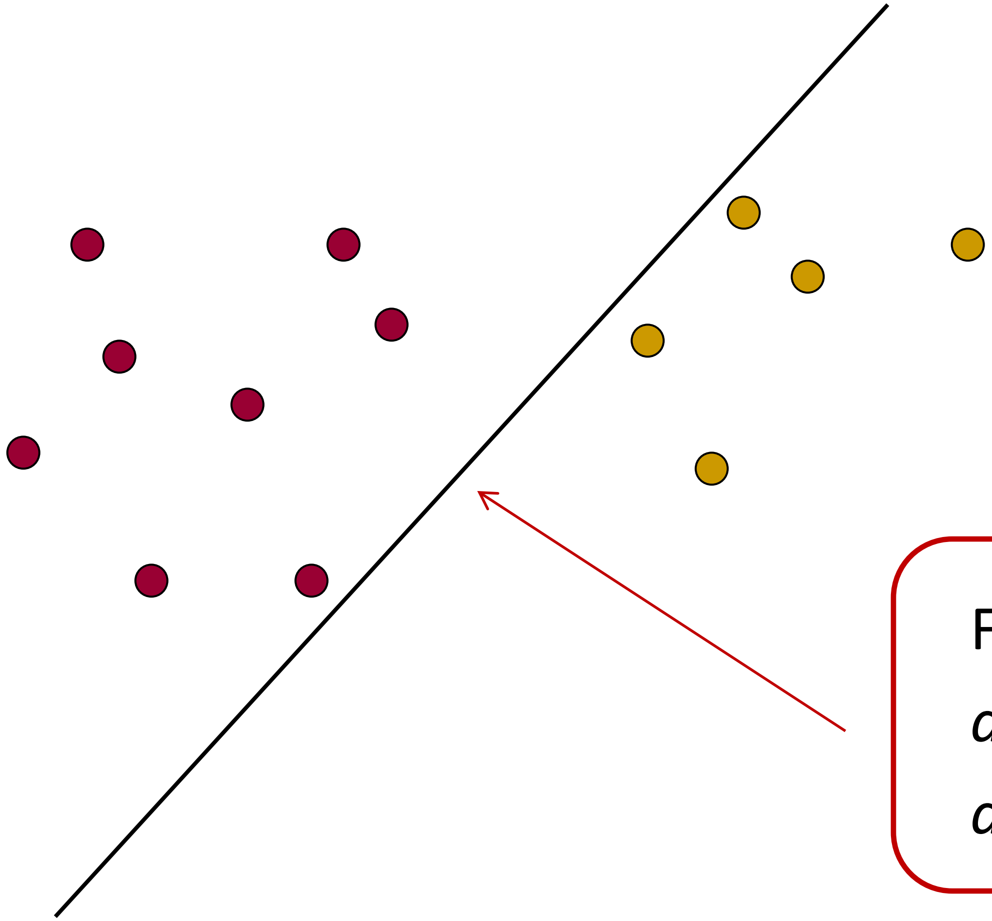
Separation by Hyperplanes

- A strong assumption is *linear separability*:
  - in 2 dimensions, can separate classes by a line
  - in higher dimensions, need hyperplanes
- Can find separating hyperplane by *linear programming*

  (or can iteratively fit solution via perceptron):
  - separator can be expressed as $ax + by = c$



Perceptron

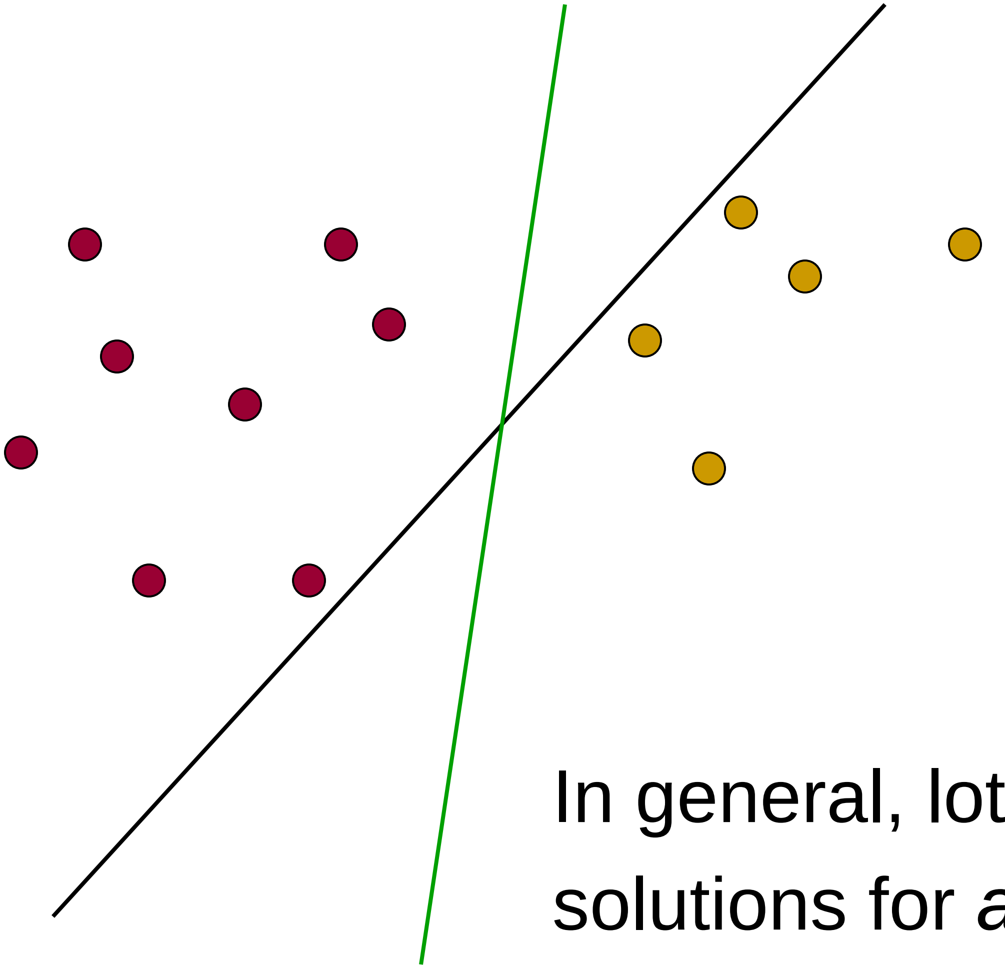


Find a,b,c, such that

$ax + by > c$ for red points

$ax + by < c$ for blue points

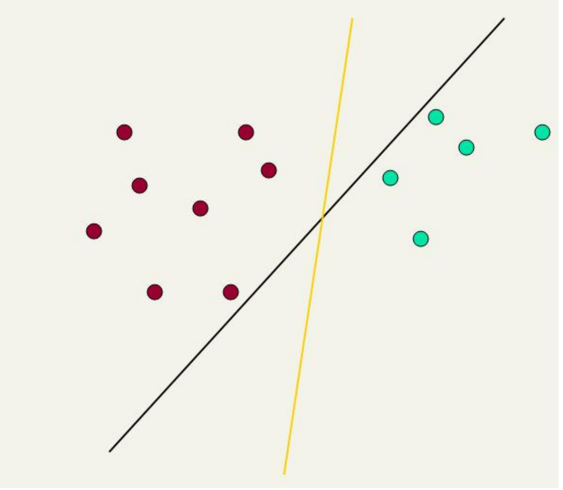
Which Hyperplane?



In general, lots of possible solutions for *a,b,c.*

Which Hyperplane?

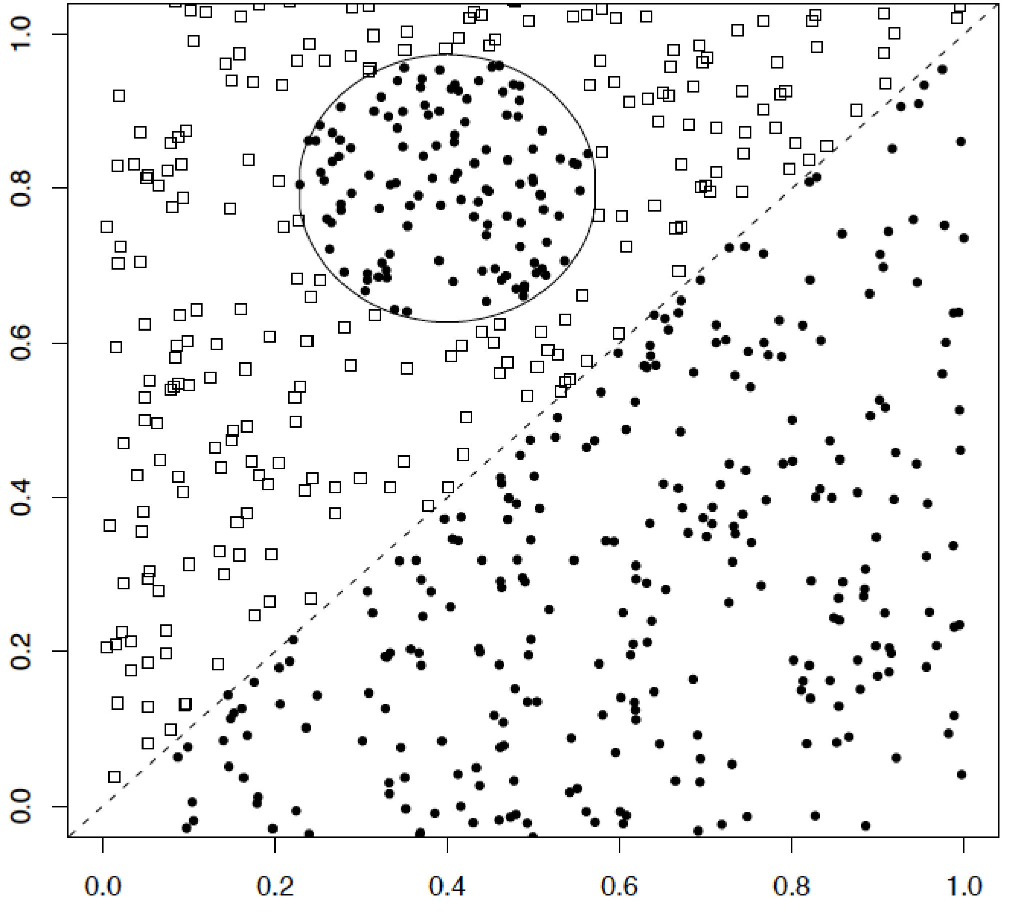
- Lots of possible solutions for *a,b,c.*
- Some methods find a separating hyperplane, but not the optimal one
  - E.g., perceptron
- Most methods find an optimal separating hyperplane [according to some criterion]
- Which points should influence optimality?
  - All points
    - Linear/logistic regression
    - Naïve Bayes
  - Only “difficult points” close to decision boundary
    - Support vector machines



Linear Classifiers

- Many common text classifiers are linear classifiers
  - Naïve Bayes
  - Perceptron
  - Centroid method
  - Linear regression / Logistic regression
  - Support vector machines (with linear kernel)
  - kNN is not linear classifier
- Despite being linear, noticeable performance differences
  - For separable problems, there is infinite number of separating hyperplanes. Which one do you choose?
  - What to do for non-separable problems?
  - Different training methods pick different hyperplanes

A nonlinear problem



- A linear method does poorly on this dataset
- kNN does well

Linear Classification can only separate space into 2 classes

How to do multi-class classification ($k > 2$)?

Use 2-class classifier to do k-class classification

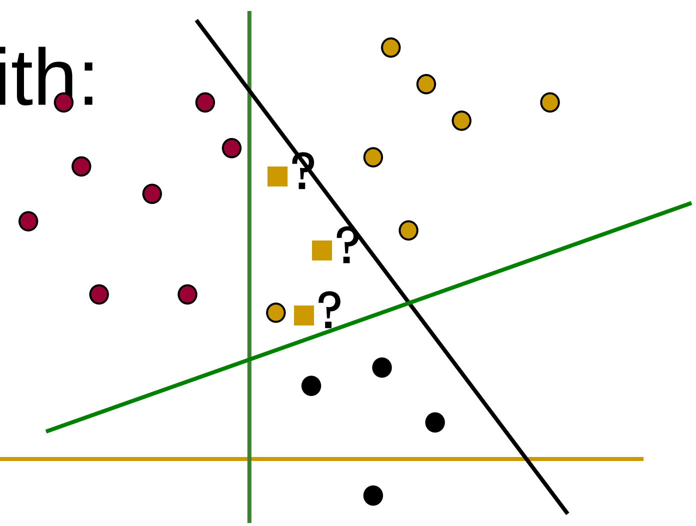
- One vs others
  - Build a classifier for each class against all other class combined together
  - Need to train K such classifiers
  - Use the largest score to determine final class
- One vs one
  - Train K(K-1)/2 classifers, each classifer one class vs another class.
  - Use majority voting to obtain final class

Multi-class Class labels

- Classes are mutually exclusive
  - Each handwritten letter belongs to exactly one class
  - A student is either 1^{st} year, 2^{nd} year, 3^{rd} year, 4^{th} year student, can not be bother or more
  - The common case: multi-class exclusive classification
- Classes are mutually non-exclusive
  - An article on drug design could also discuss the drug company’s (and market) economics.
  - An image has sky, building, road etc.
  - Multi-class inclusive classification (multi-label classification)

One vs Others: more details

- Build a classifier between each class and its complementary set (docs from all other classes).
- Given test object, evaluate it for membership in each class.
- Assign document to class with:
  - maximum score
  - maximum confidence
  - maximum probability



High Dimensional Data

- Pictures like the one at right are absolutely misleading!
- Documents are zero along almost all axes
- Most document pairs are very far apart (i.e., not strictly orthogonal, but only share very common words and a few scattered others)
- In classification terms: often document sets are separable, for most any classification
- This is part of why linear classifiers are quite successful in this domain