CIS501 – Lecture 5

Woon Wei Lee
Fall 2013, 10:00pm-11:15pm,
Sundays and Wednesdays

For today:

- Announcements:
  - Textbooks!
  - Make-up arrangements for next week..
- Bayes decision rule
- *k*-NN classifier
- Presentations
  - Aishah Al Yammahi
  - Ioannis Karakatsanis

(Refresher) **Bayes Theorem**

- Bayes theorem is given by:

$$p(c|x) = \frac{p(c,x)}{p(x)}$$

$$= \frac{p(x|c)\,p(c)}{p(x)}$$

- Specialized terms in Bayesian Analysis:
  - *c* - The model or property to be inferred
  - *x* -The “observations”
  - *p(x|c)* – The “Likelihood”
  - *p(c)* – The “prior”
  - *p(c|x)* – The “posterior”
  - *p(x)* – The “evidence”

Thomas Bayes
1702-1761

Masdar INSTITUTE

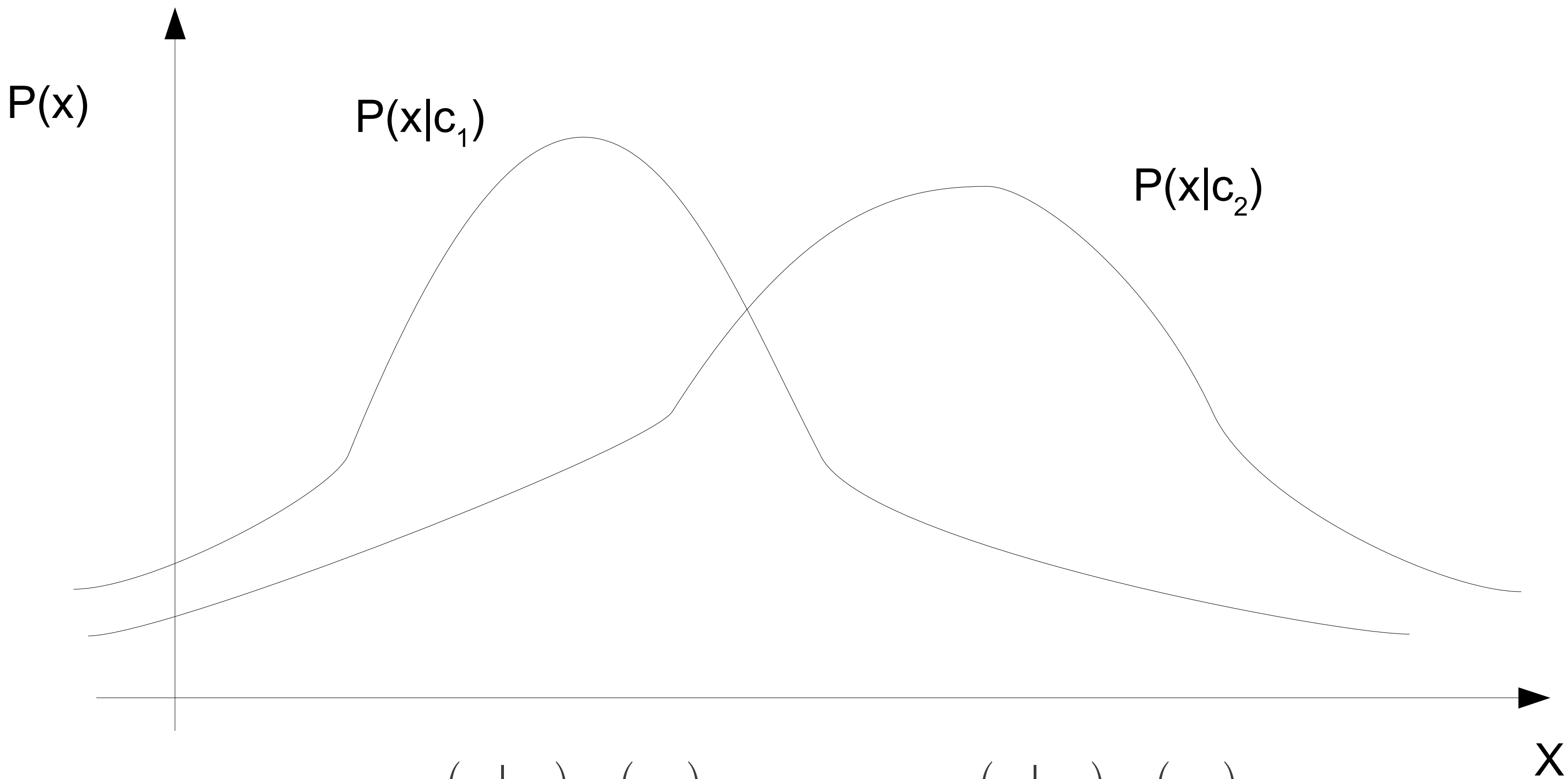
Bayes decision rule

- In the case of classification, "*c*" denotes the category or class from which the data was sampled
- In general, the classification problem is as follows:
  - Given a particular observation, *x* and *n* potential classes, determine the class which satisfies:

$$c = \underset{\forall i \in \{1,2,\ldots,n\}}{argmax_i}\, p(c_i|x)$$

- *p(x)* is independent of the class, and..
- .. *p(c)* is frequently assumed to be the same for all classes.
- In which case, the likelihood term is interchangeable with the posterior term above.

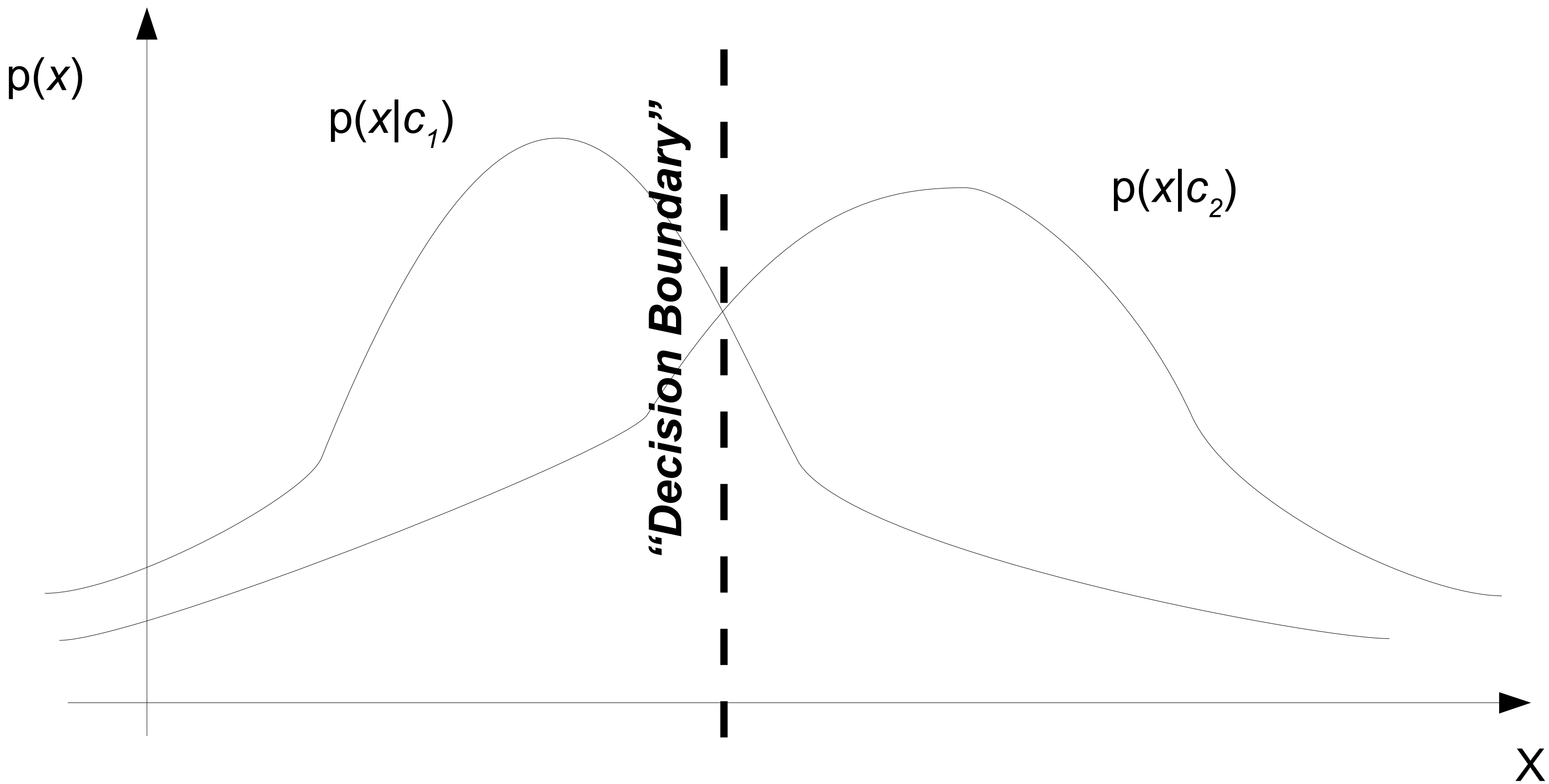
2-class case



$$P(c_1|x) = \frac{p(x|c_1)\,p(c_1)}{p(x)} = \frac{p(x|c_1)\,p(c_1)}{p(x|c_1)\,p(c_1) + p(x|c_2)\,p(c_2)}$$

$$P(c_2|x) = \frac{p(x|c_2)\,p(c_2)}{p(x)} = \frac{p(x|c_2)\,p(c_2)}{p(x|c_1)\,p(c_1) + p(x|c_2)\,p(c_2)}$$

2-class case: p(c_1)=p(c_2)



$$P(c_1|x)=\frac{p(x|c_1)\,p(c_1)}{p(x)}=\frac{k_1\,p(x|c_1)}{k_2}$$

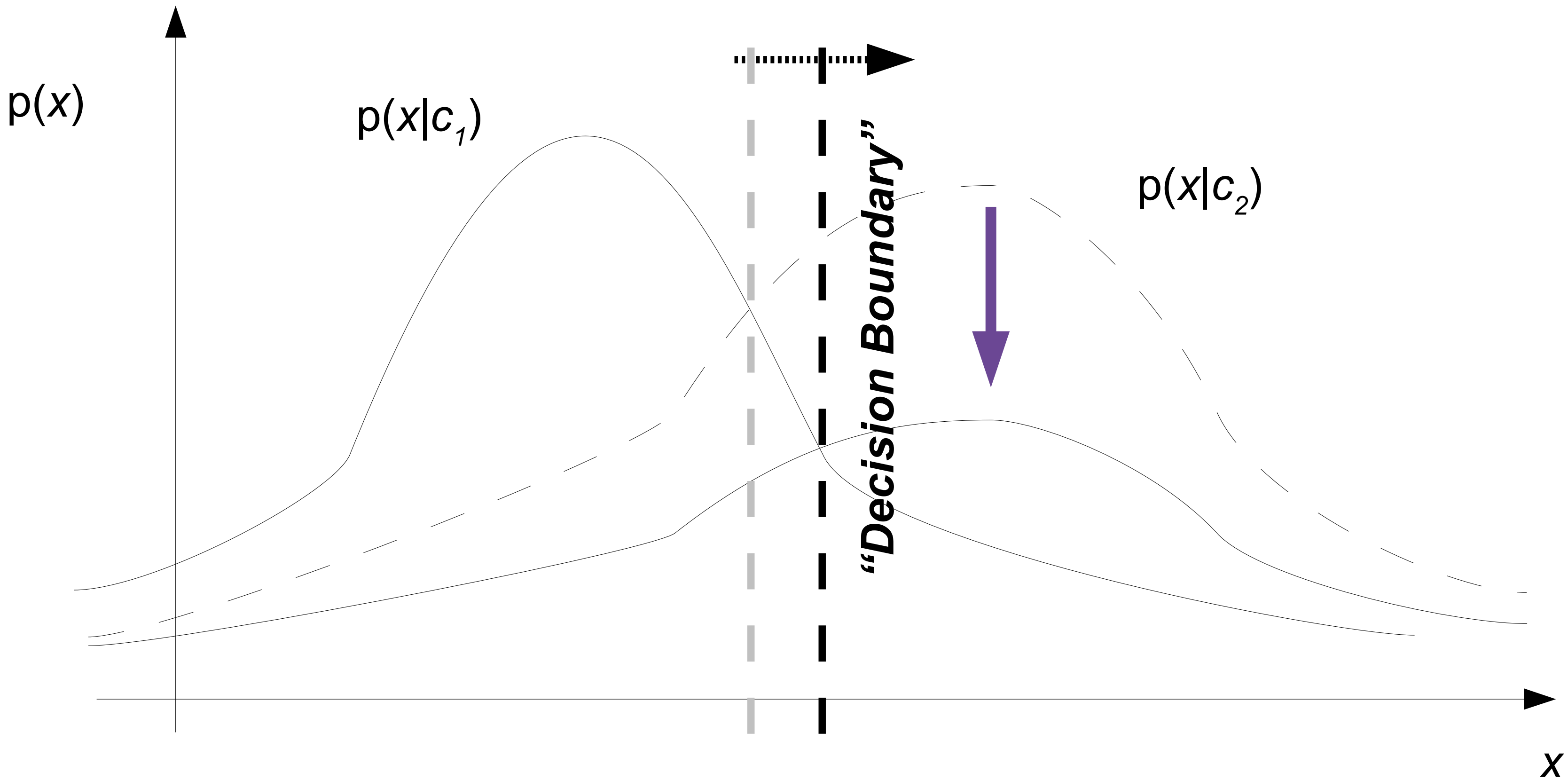
$$P(c_2|x)=\frac{p(x|c_2)\,p(c_2)}{p(x)}=\frac{k_1\,p(x|c_2)}{k_2}$$

Decision boundary occurs when:

p(x|c₁)=p(x|c₂)

Instance of "maximum likelihood" classification

2-class case: $p(c_1)=2 \times p(c_2)$



$$P(c_1|x)=\frac{p(x|c_1)\,p(c_1)}{p(x)}=\frac{2\mathrm{k}_1\,p(x|c_1)}{k_2}$$

$$P(c_2|x)=\frac{p(x|c_2)\,p(c_2)}{p(x)}=\frac{k_1\,p(x|c_2)}{k_2}$$

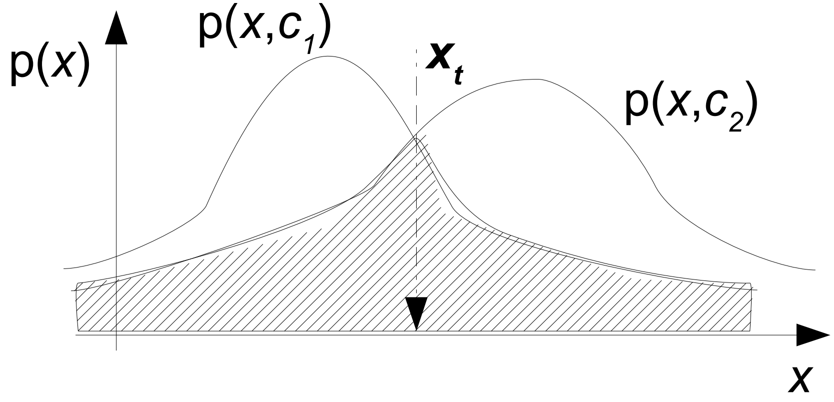
Classification as risk analysis

- What is the probability of misclassification?

$$p(mistake) = p(x > x_t, c_1) + p(x < x_t, c_2)$$
$$= p(x > x_t | c_1)\, p(c_1) + p(x < x_t | c_2)\, p(c_2) \quad .... \quad (1)$$

- This is the shaded area under the curve
- **Risk analysis perspective**
  - "Bayesian risk" - defined as "expected value of loss":

$$Risk = E[L] = \int L(y)\, p(y)\, dy$$
$$= \int \int L(x, c)\, p(x, c)\, dx dc$$
$$= \sum_{c \in 1,2} \int L(x, c)\, p(x|c)\, p(c)\, dx$$



- **Generalizes misclassification probability:**
  - If loss function is 1 for all misclassified cases, then we just get (1)
  - Possible to have different loss functions

Question: examples?

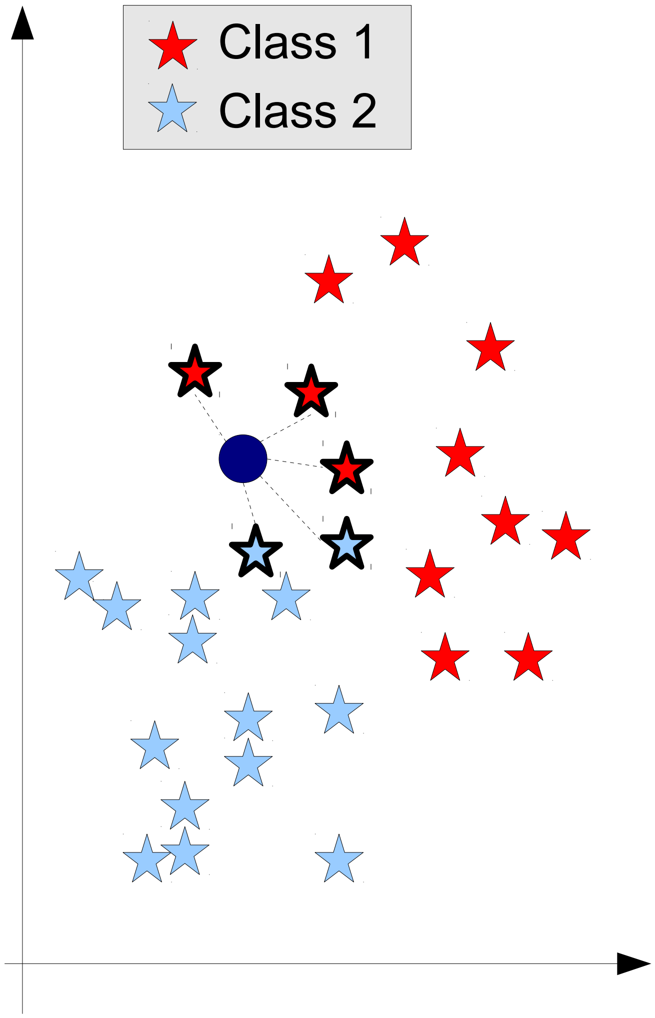
Bayesian Formulation (cont'd)

- **Disadvantages:**
  - More tedious
  - Often requires various assumptions/approximations to obtain values for various parameters → often unfounded
- **Advantages:**
  - More "principled" - motivates Bayes decision rule.
  - Allows confidence in the results to be estimated naturally
  - Most of the assumptions/approximations are made implicitly anyway.
  - Knowing the probability allows for techniques like:
    - Bayesian Risk framework
    - Creation of ROC charts (to be covered later)

The *kNN* classifier - our first (real) classifier!

- We can use exactly this expression to build a "kNN" classifier, i.e.:
  - For each *x* to be classified, estimate $p(x|c_1)$ and $p(x|c_2)$ separately
  - Calculate $p(c|x)$ and apply Bayes decision rule
- However, in the "standard" version of *k*NN, the following simplified procedure is applied:
  1. For each *x* to be classified, find the nearest *k* training instances of any class.
  2. Determine the number of instances of class 1 and 2, let's say n_1 and n_2
  3. Assume that $p(c_1)=p(c_2)$
  4. In which case, maximum likelihood rule can be used, with:
     - $p(x|c_1) \propto n_1$ and $p(x|c_2) \propto n_2$

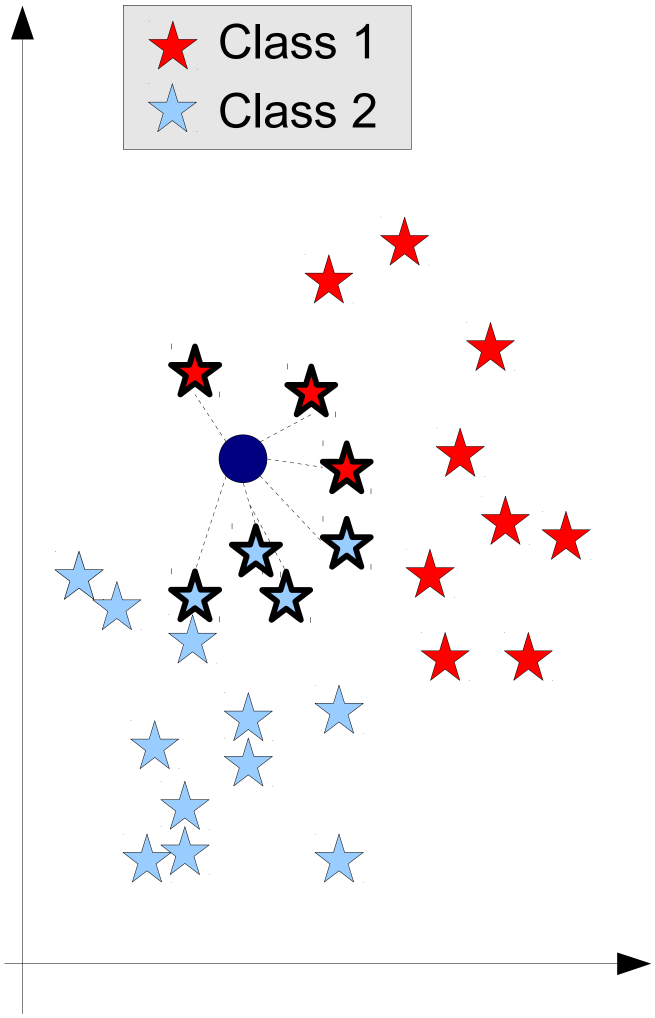
The *kNN* classifier (cont'd)



- For e.g.: in example on left:
  - k=5
  - $n_1=3, n_2=2$

  $\Rightarrow$ Unseen example is classified as class 1

The *kNN* classifier (cont'd)

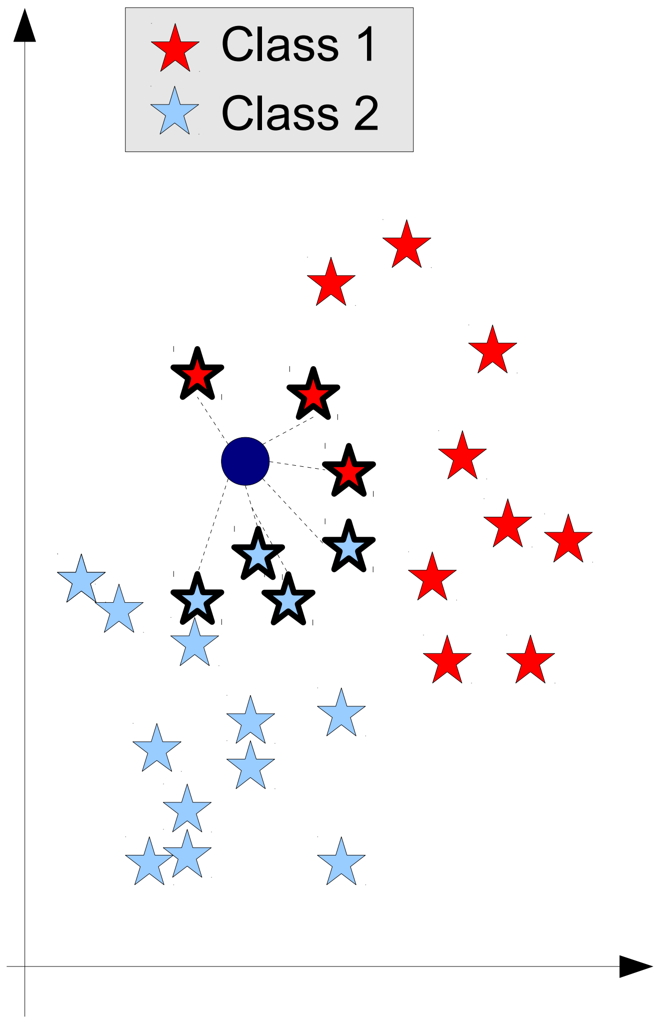


- For e.g.: in example on left:
  - k=5
  - $n_1=3,\ n_2=2$

    ⇒ Unseen example is classified as class 1
  - k=7
  - $n_1=3,\ n_2=4$

    ⇒ Unseen example is classified as class 2

The *kNN* classifier (cont'd)

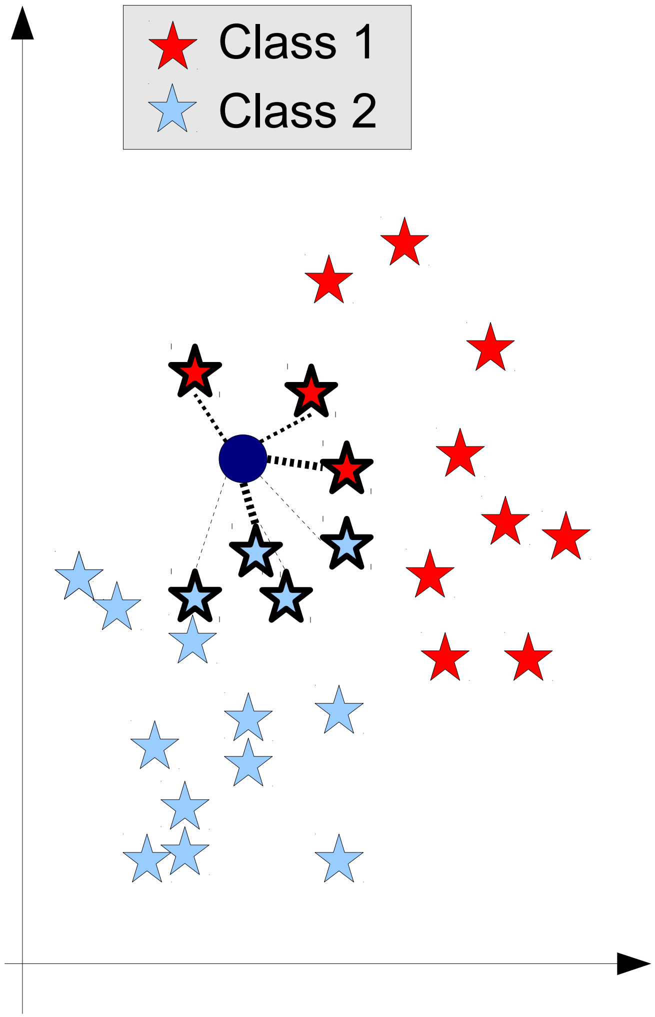


- For e.g.: in example on left:
  - k=5
  - $n_1=3, n_2=2$

    ⇒ Unseen example is classified as class 1
  - k=7
  - $n_1=3, n_2=4$

    ⇒ Unseen example is classified as class 2
- Two extreme examples:
  - $k=1$ ⇒ Just pick closest training example
  - $k=n$ ⇒ ***Automatically*** choose the most common class

Distance weighted k-NN classifier



- Standard *k*-NN:
  - Big *k* → good noise resistance, poor resolution
  - Small *k* (the opposite)
- One trick is to emphasize closer neighbors:
  - Assign different weightings to the neighbors
  - Different weighting schemes available have been suggested:

i. $$w_j = \frac{d_k - d_j}{d_k - d_1}$$

ii. $$w_j = \frac{1}{d_j}$$

iii. $$w_j = k - j + 1$$

A useful generalization..

- A further "advantage" of the kNN → strictly speaking, only the distance between points is required.
- So far, the distance function that we have been assuming is the "Euclidean distance"

  i.e. $d(v_1,v_2)=\sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$

- In principle however, we can apply kNN using any kind of distance!

  1. Set intersections
  2. Edit distances
  3. Angular differences
  4. Manhattan distance
  5. Mahalanobis distance