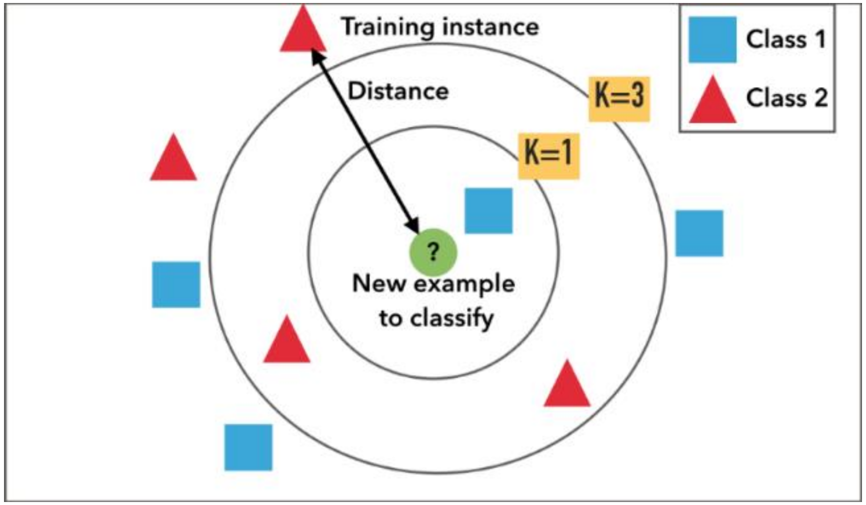
Chemical Engineering 4H03

Nearest Neighbours Classification

Jake Nease
McMaster University



https://medium.com/@anuuz.soni/advantages-and-disadvantages-of-knn-ee06599b9336

Objectives for this Topic

- We would like to introduce a new form of classifiers based on **supervised learning**
  - This will be a **short lesson** outlining the basics (there is not much to it, to be honest)
- Introduction to Nearest Neighbours
  - How to use
  - Neighbour classifier versus radial classifier
  - Benefits and pitfalls
- Introduction to nearest centroids
- Quick example in MATLAB

NN In a Nutshell

- Neighbour-based classification is **instance-based**
  - Does not attempt to generate an internal model
  - Thus new points are not classified in the typical $\hat{y}$ fashion

- Instead, NN stores instances of the training data as a comparative array
  - Can be updated as new data is (correctly) classified in the future

- Classification is quite simple: it is compute as a **majority vote** of the nearest neighbours of a query point
  - The query point is assigned to whichever class holds the most votes in the neighbourhood

Computing K-NN Classifications

- Consider a set of training data:

$$\mathcal{T} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_i, y_i), \cdots, (\boldsymbol{x}_N, y_N)\}$$

- Each point $\boldsymbol{x}_i \in \mathbb{R}^{K\times 1}$ is a vector of independent data in a K-dimensional space belonging to class y_i
  - Here, y_i could belong to a set $\{1,2,3 \ldots\}$ or $\{\mathrm{Y}, \mathrm{N}\}$ or $\{\mathrm{GOOD}, \mathrm{BAD}, \mathrm{UGLY}\}$ depending on the context
  - The set of possible y_i must be known *a-priori* (note this is **different than K-Means**, which did not know the class to which the data belonged beforehand*)

- Of special note: all entries $\boldsymbol{x}_i$ must be continuous values (no names, strings… binaries OK but not great)

*Nor after, technically… But it thinks it does

Computing NN Classifications

- The easiest way to classify a query point $\tilde{\boldsymbol{x}}$ (with same dimensions as all $\boldsymbol{x} \in \mathcal{T}$) is to use the **brute-force method** as follows:

1. Select the number of nearest-neighbours $\mathcal{N}$ required for voting (must be decided before classification)
2. Compute the distance from $\tilde{\boldsymbol{x}}$ to $\boldsymbol{x}_i$ for all i as:

$$d_i = \|\tilde{\boldsymbol{x}} - \boldsymbol{x}_i\| \;\; \forall\, i$$

3. Select the $\mathcal{N}$ shortest distances and corresponding points $\boldsymbol{x}_i$. Place $\boldsymbol{x}_i$ in the "voting set" $\mathcal{V}$
4. Query $\tilde{\boldsymbol{x}}$ belongs to class y_m, where y_m is the classifier corresponding to the majority votes from $\boldsymbol{x}_i \in \mathcal{V}$

Comments

- This is definitely the easiest way to go

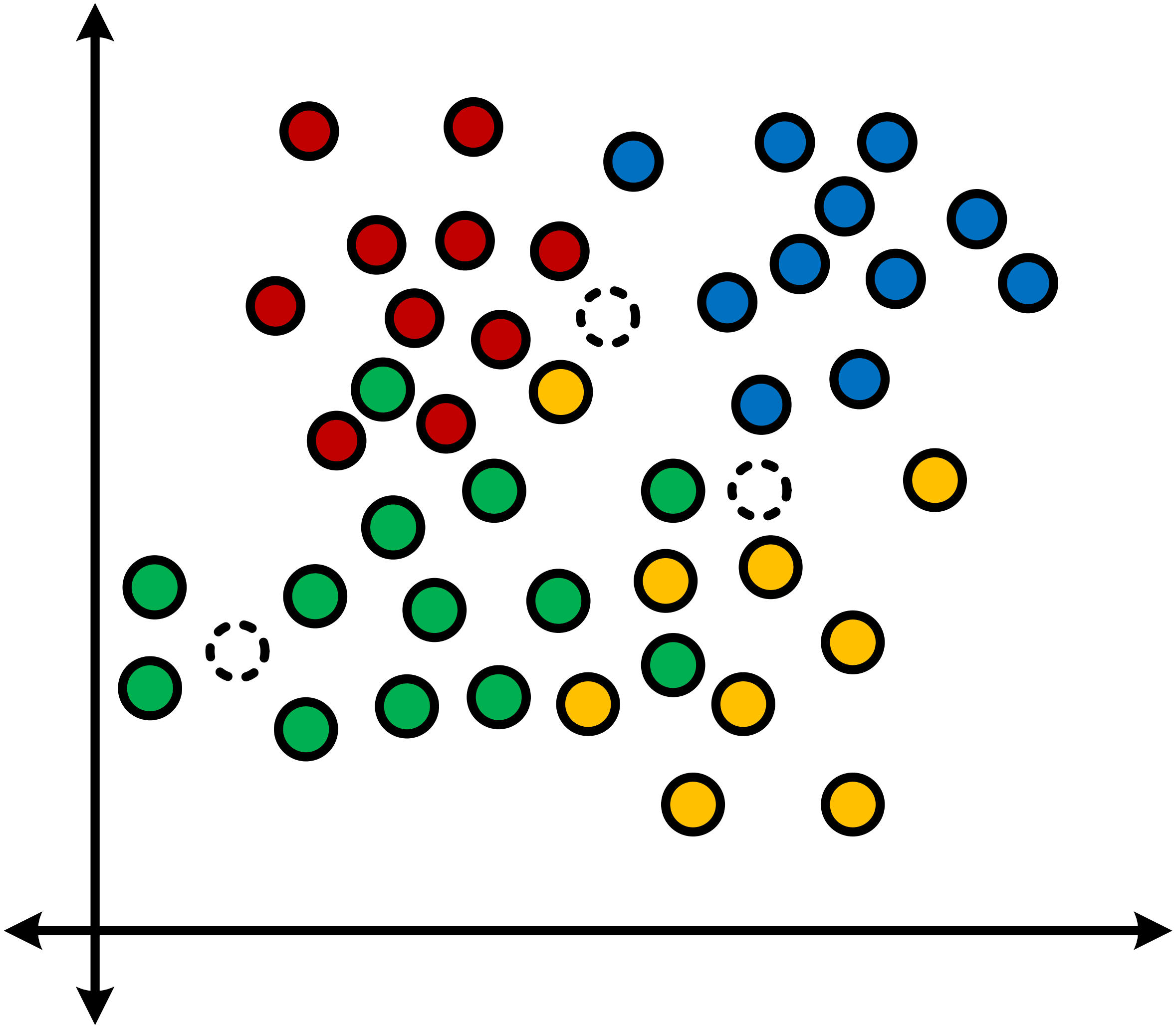
- Choosing $\mathcal{N}$ depends highly on the data
  - Higher $\mathcal{N}$ tends to eliminate noise
  - However, higher $\mathcal{N}$ makes potential delineation between subsets less distinct

- Basic algorithm treats all points equally
  - Possible to weight different points to contribute more or less to the classification strength (*i.e.* higher weights for points closer to the query point)

- Able to classify points in data sets that overlap
  - Added information of which point belongs in which class helps here

Visual Example

- How would you classify the three query points?



Radial NN Classifications

- Very similar to NN classification, but with a twist

- Instead of choosing $\mathcal{N}$ nearest neighbours, the user reports a radius r
  - Can be chosen by-dimension (leading to a hyper-ellipse of sorts) but is often not done that way

- All points within hypersphere of radius r are included in the voting set $\mathcal{V}$

[*] Nor after, technically… But it thinks it does

Shortcomings of NN

- Requires the selection of $\mathcal{N}$ neighbours up front
  - Results may depend strongly on how many neighbours are used

- As the total number of points $N \rightarrow \infty$, NN becomes more accurate but suffers from long computation time

- As $K \gg 1$, randomly drawn points from a probabilistic distribution (the backbone of NN) become less and less similar to each other, leading to poor predictive power
  - However, if the data in **X** behave in correlated sub-spaces, we can use methods like PCA to dimensionally reduce the data OR use nonlinear partitioning methods like SVMs

- Is technically a "supervised learning" method, and thus not helpful if we don't know training classes ahead of time

Benefits of NN

- Very easy to use, relatively speaking
- Allows for classification of data with nonlinear separation (even if we require the true outcomes via the training set)
- Allows for classification of data sets with overlap
  - This is where that probabilistic distribution thing comes in!

What Will be Our Last Topic?

- DECISION TREES and RANDOM FORESTS
  - Used to classify binary outcomes based on entropy minimizations of data sets
  - Supervised learning

- SUPPORT VECTOR MACHINES
  - Fitting a kernel to a set of data that delineates between known classes

- Remaining Course Schedule
  - Wrap up and overview April 06
  - Mahir’s ANN applications to research Apr 08
  - Presentations following!