

k Nearest Neighbours Algorithm

Given:

- similarity metric (like cosine),
- parameter k ,
- reference set $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m\}$,
- query \vec{y} ,
- target classes, c_1, \dots, c_d , and assignments to X .

1. Initialise $L(c_j) := 0$ for each class j
2. For all $\vec{x}_i \in k$ closest training vectors to \vec{y} :
 - For each class c_{ji} to which \vec{x}_i belongs:
 - $L(c_{ji}) += \text{sim}(\vec{x}_i, \vec{y})$
3. Choose c_j with largest $L(c_j)$

k Nearest Neighbours Algorithm

Advantages:

- no training phase,
- guaranteed error bounds (with enough data):
when $k=1$, it converges to $2 \times$ *Bayes error rate*,
the optimal error rate attainable by maximising
 $P(c_j | \vec{y}, X)$.
- fairer weighting of evidence than cosine.

Disadvantages:

- must choose k ,
- must choose similarity metric,
- time/space complexity not good: $\mathcal{O}(n \cdot |X|)$
(n = dimension of \vec{y} and \vec{x}_i),
- performance not good if variances of classes are different.

But there are fast approximations: choose k that are pretty close, but perhaps not closest (ϵ -NN).