Classification Problems

Examples:
- Fraud detection in credit card payments
- Categorize digital images : dog, cat, bus, house ....

- Supervised learning: Similar to regression problems,
  - $X \subset \mathbb{R}^k$: domain set
  - $Y \subset N$: finite target set, $|N| = q$

1. (Multiclass) Logistic Regression.

$\mathcal{X} \subset \mathbb{R}^k$, $\mathcal{Y} = \{0,1\}^q$, $\mathcal{F} = \{ x \mapsto \text{softmax}(Wx), W \in \mathbb{R}^{q \times k} \}$ $\mathbb{R}^k \to \mathbb{R}^q$,

with $\text{softmax}(z) = \left[\frac{\exp(z_1)}{\sum_{i=1}^q \exp(z_i)} \cdots \frac{\exp(z_q)}{\sum_{i=1}^q \exp(z_q)} \right]$,

- Loss fanction $\ell(y, z) := -\sum_{i=1}^q y_i \log(z_i) = -\langle y, \log(z) \rangle$

- empirical risk $L_S(W) = \frac{1}{n} \sum_{j=1}^n \ell(y^j, \text{softmax}(Wx^j))$,

with hot encoding s.t. $y = (0,0,\dots,0,1,\dots,0)$ ← if data point in i-th class.

- Adding ridge/Lasso like regularization term is possible.
- Optimization is non-trivial, but doable since $W \mapsto L_S(W)$ is convex.

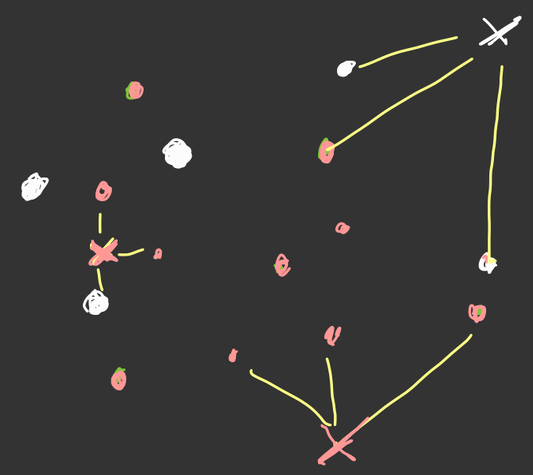
2. K-Nearest Neighbors classification:

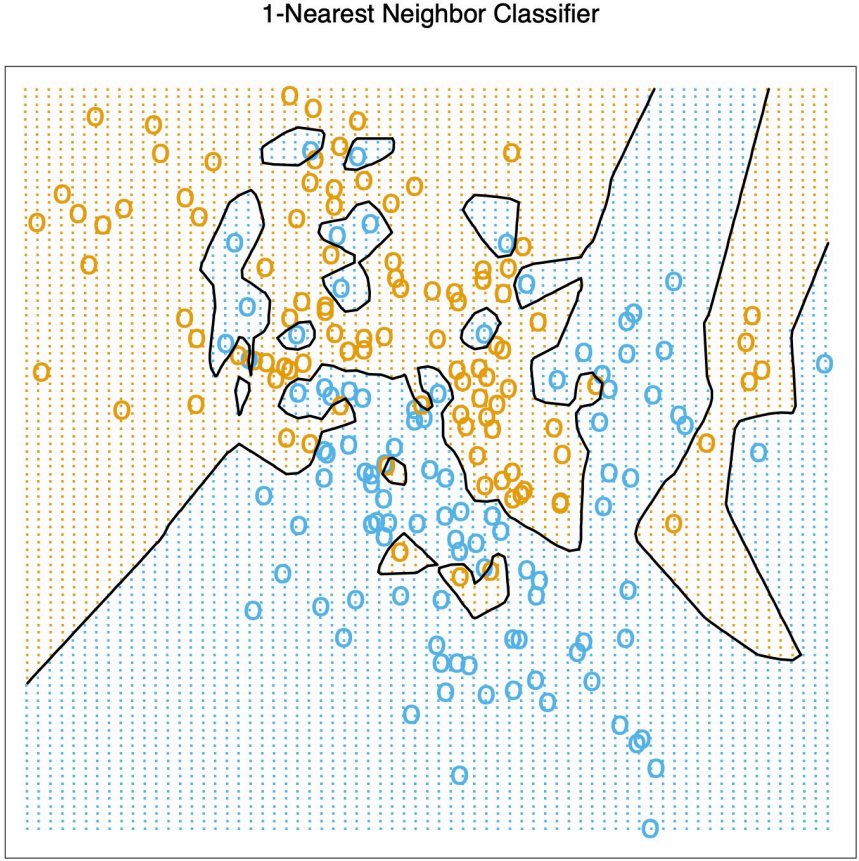
Let $X \subset \mathbb{R}^n$, $Y \subset \{0,1\}^q$. Let $d: X \times X \to \mathbb{R}$ be a metric, e.g., $d(x,x') = \|x - x'\|_2$.
If $S_x = \{x_1, \dots, x_n\}$ is a set, define $\pi_i(x)$ as the i-th closest member of S_x to x with respect to d.

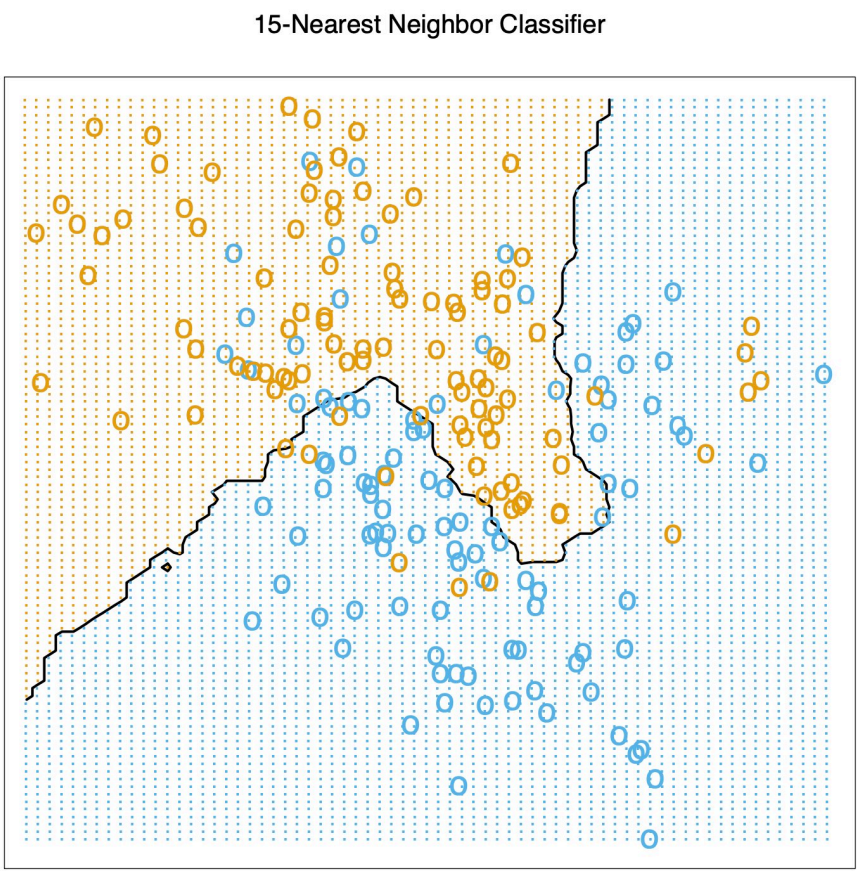
Algorithm: Input: Training set $S = (x_i, y_i)_{i=1}^n$, parameter k.
Output: Function $h_S: X \to Y$ such that $h_S(x)$ is the majority label among $\{y_{\pi_i(x)} : i \le k\}$.

- ⊕ "local" method, simple
- ⊖ Needs all pairwise comparisons ($O(kn)$ computations*)
- ⊖ Suffers from "curse of dimensionality" if d large.

Choose $k=3$.




1-Nearest Neighbor Classifier


15-Nearest Neighbor Classifier

Feature selection: Natural language Processing

Q: Use studied approaches for text data?

→ spam/no spam classification

→ categorizing text items according to topics

$x =$ " Michael likes walking his dog in his neighborhood. "

1. Tokenize:

"Michael", "likes", "walking", "dog", "in", "his", "neighborhood"

2. Build vocabulary: Do this for a lot of documents:

"Aaron", "Amsterdam", "am", ..., "his", ..., "Michael", ..., "walking", ..., "zebra", ... List of k words.

3. Encoding:

a) Count how often each word in x occurs, create sparse vector in $\mathbb{R}^k$ (normalized):

"Count encoding"

$$\Phi : \mathcal{D} \longrightarrow \mathbb{R}^k$$

$$\Phi(x) = \frac{\tilde{\Phi}(x)}{\|\tilde{\Phi}(x)\|}$$

with $\tilde{\Phi}(x) = (0, 0, 0, \ldots, 2, \ldots, 1, \ldots, 1, \ldots, 0)$

Modifications can include:

- ▷ Removal of common words such as "in" "the"
- ▷ "n-grams": Use "Michael likes", "likes walking", etc. as words.
  - ⊕ Better semantic understanding
  - ⊖ Compatationally more challenging as dictionary larger

b) Term Frequency - Inverse Document Frequency (TF-IDF)

Choose $\Phi(x)_w = freq_w \cdot \left(\log\left(\frac{K}{N_w}\right) + 1\right)$

$freq_w$: fregency of word w in document x

K: nr. of words

N_w: nr. of documents containing word w.

⊕ scales down importance of word common across documents.