

# CS 221 Section 2: Learning

Chuma Kabaghe

# Nearest neighbors



## Algorithm: nearest neighbors

Training: just store  $\mathcal{D}_{\text{train}}$

Predictor  $f(x')$ :

- Find  $(x, y) \in \mathcal{D}_{\text{train}}$  where  $\|\phi(x) - \phi(x')\|$  is smallest
- Return  $y$

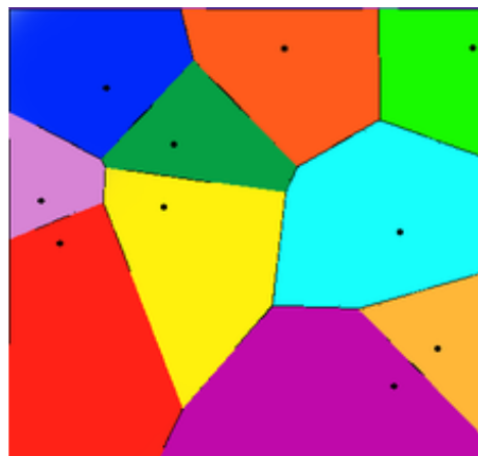


## Key idea: similarity

Similar examples tend to have similar outputs.

# Expressivity of nearest neighbors

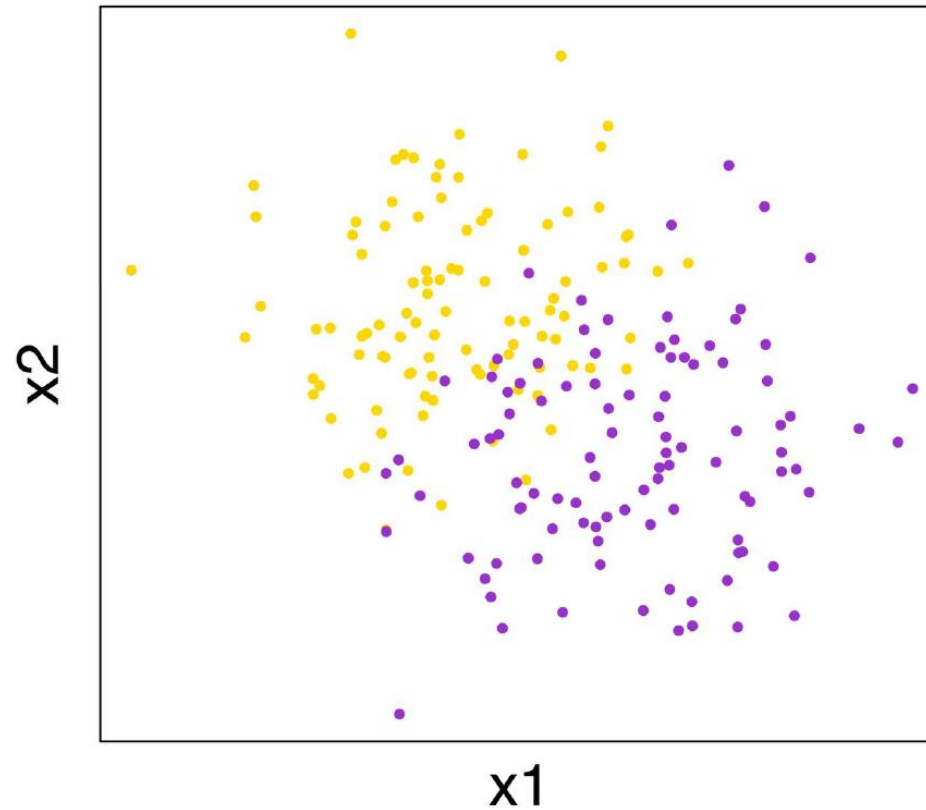
Decision boundary: based on Voronoi diagram



- Much more expressive than quadratic features
- **Non-parametric**: the hypothesis class adapts to number of examples
- Simple and powerful, but kind of brute force

# *k*-nearest neighbors

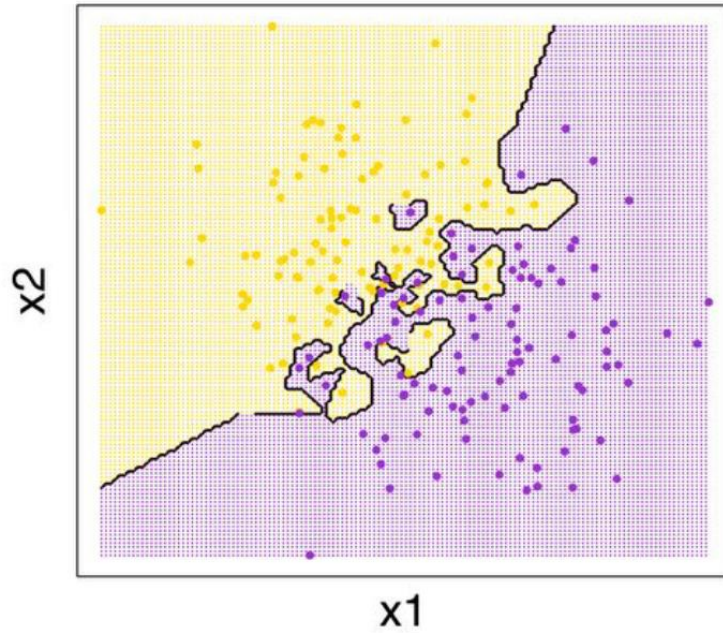
**Binary kNN Classification Training Set**



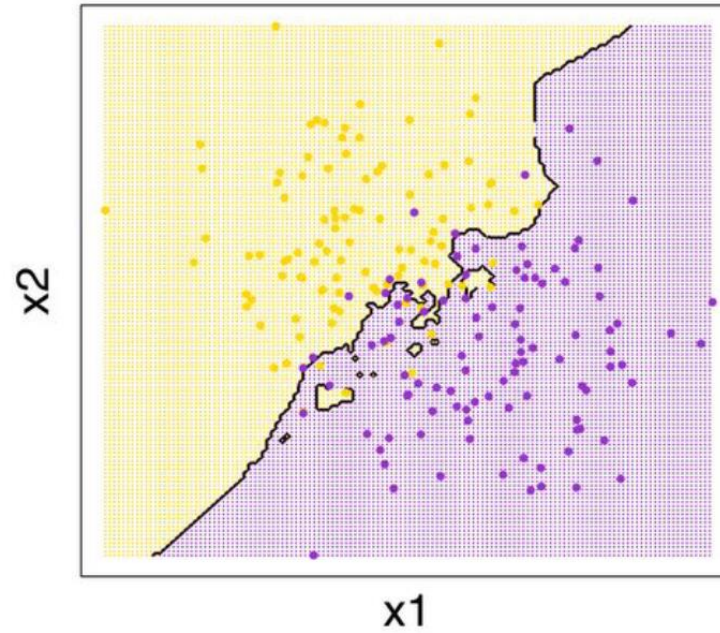
# $k$ -nearest neighbors

Effect of  $k$ :

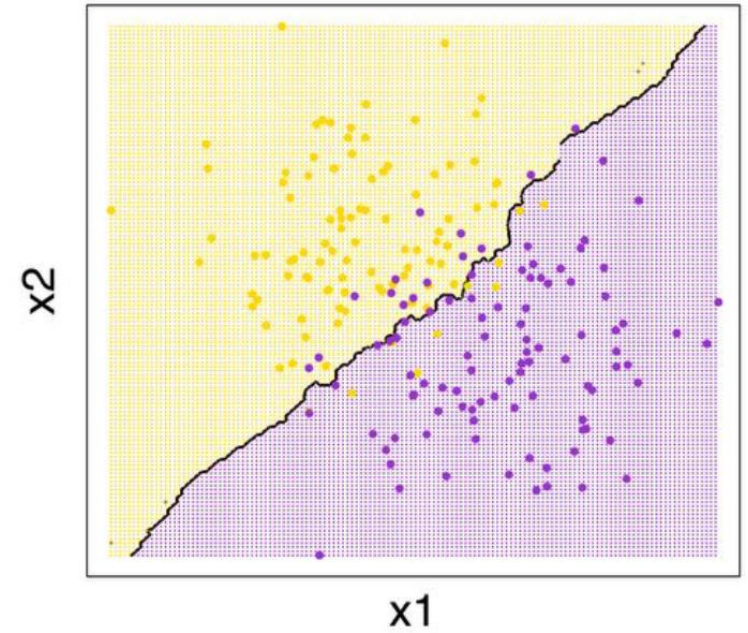
Binary kNN Classification ( $k=1$ )



Binary kNN Classification ( $k=5$ )



Binary kNN Classification ( $k=25$ )



[Switch to ipython notebook]

# $k$ -fold cross-validation

