k-Nearest Neighbour

Gautam Kamath

Bayes Classifier

- Recall: goal in classification is to solve $\min_f \Pr_{x,y\sim D}[f(x) \neq y]$
  - Comment: we previously talked about *loss functions*, not this classification error. Classification error is harder to optimize, so we use loss fns as a proxy
- Bayes optimal classifier: $f^*(x) = \arg\max_c \Pr_{y\sim D_{Y|X=x}}[y = c|x]$
  - Given a point x, look at the distribution of labels given that feature vector
  - Pick whichever label is most likely to be generated
  - Caveat: requires knowing the data distribution D, in general impossible
- No classifier can ever do better
- (Draw examples: where labeled 1 for $x \in S$ 0 otherwise, with linear classifier and probabilities go up farther, truly random)
- Error of Bayes optimal classifier: $E_{x\sim D_X}\left[1 - \max_c \Pr_{y\sim D_{Y|X=x}}[y = c|x]\right]$

k-Nearest Neighbours

- Implicit assumption: if feature vectors x and x' are close, then labels y and y' are likely to be the same
  - $\Pr_{y \sim D_{Y|X=x}}[y = c|x] \approx \Pr_{y' \sim D_{Y|X=x'}}[y' = c|x']$ when x and x' are close

Algorithm: kNN

Input: Dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \in \mathsf{X} \times \mathsf{Y} : i = 1, \ldots, \mathsf{n}\}$, new instance $\mathbf{x} \in \mathsf{X}$, hyperparameter k

Output: $\mathbf{y} = \mathbf{y}(\mathbf{x})$

for $i = 1, 2, \ldots, \mathsf{n}$ **do**

  $d_i \leftarrow \mathrm{dist}(\mathbf{x}, \mathbf{x}_i)$ `// avoid for-loop if possible`

find indices $i_1, \ldots, i_k$ of the k smallest entries in $\mathbf{d}$

$\mathbf{y} \leftarrow \mathtt{aggregate}(\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_k})$

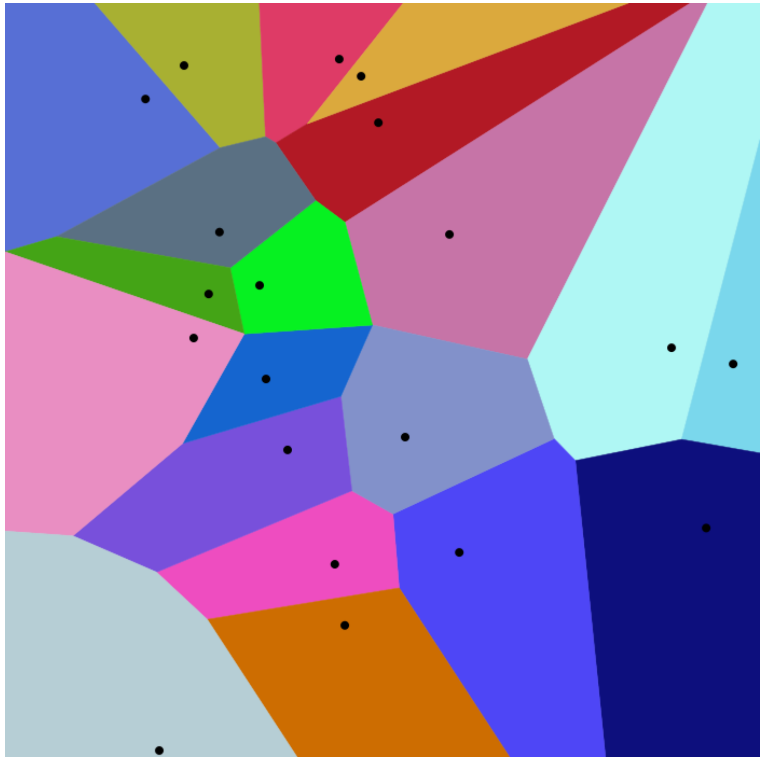
- Dist and aggregate underspecified: often ℓ_2 and majority vote
- (Draw an example, say $k = 5$)

Comments on kNN

- Non-parametric
  - Can’t be succinctly described by a parameter vector
- Distances
  - (Draw ℓ_2 ball versus ℓ_1 and ℓ_∞ ball)

Time and Space Complexity

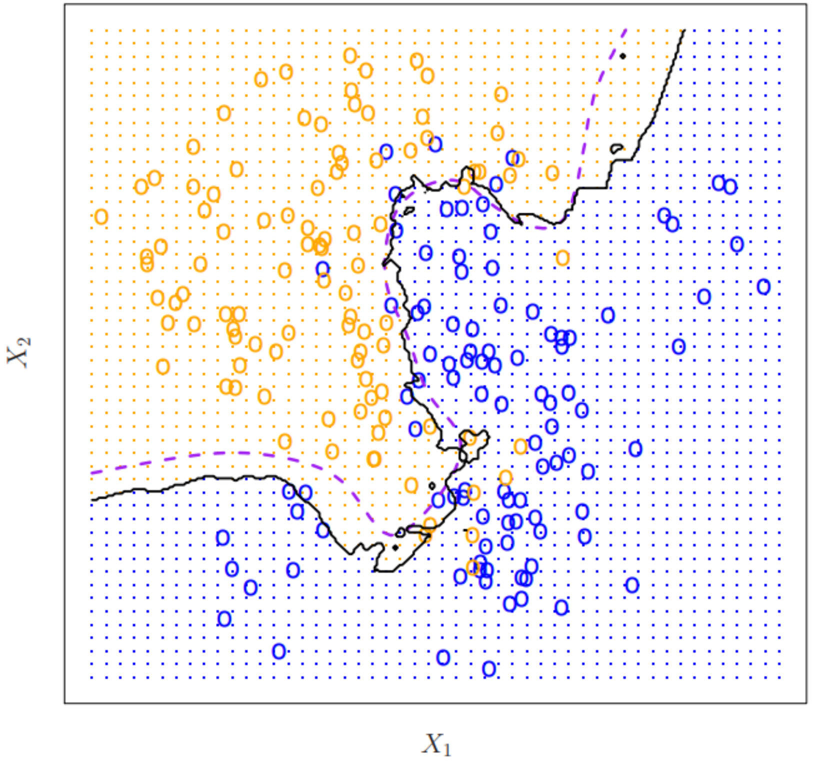
- Training takes 0 time (just store dataset), but $O(nd)$ space
  - Compare space with perceptron, lin reg, which only need $O(d)$ space
- Classification of new point takes $O(ndk)$ time naively, $O(nd)$ space
  - Time can be reduced to $O(nd)$ time a bit more carefully
- Can do better in some cases
  - E.g., Voronoi diagram for 1-NN
    - Takes $O(d \log n)$ time, $n^{O(d)}$ space
    - Good in low-dimensional settings
  - Approximate nearest neighbours



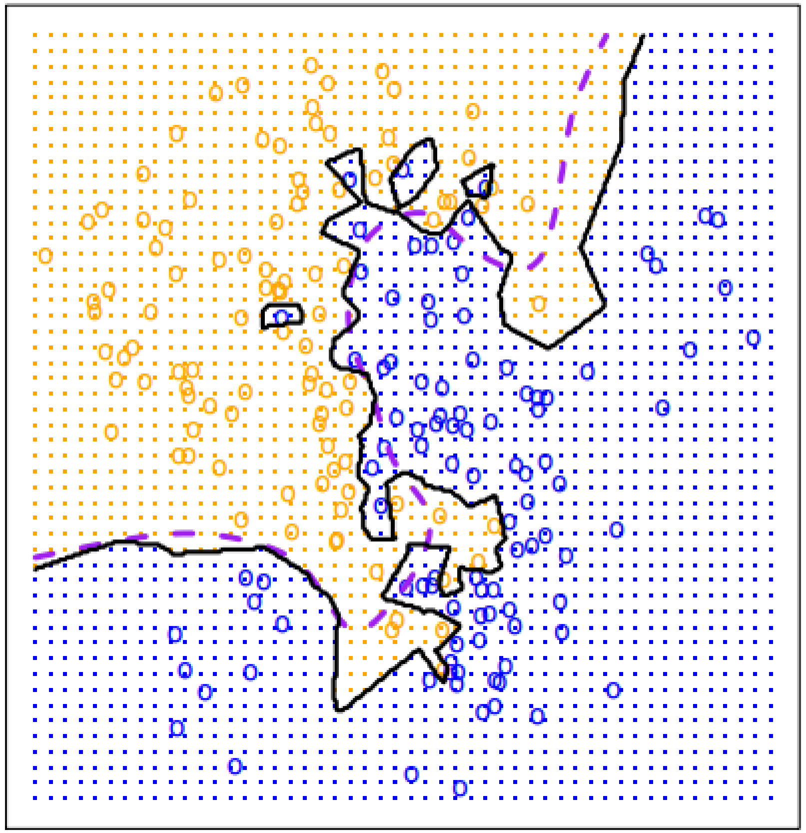
The role of k

- (Revisit previous $k = 5$ with larger and smaller values)

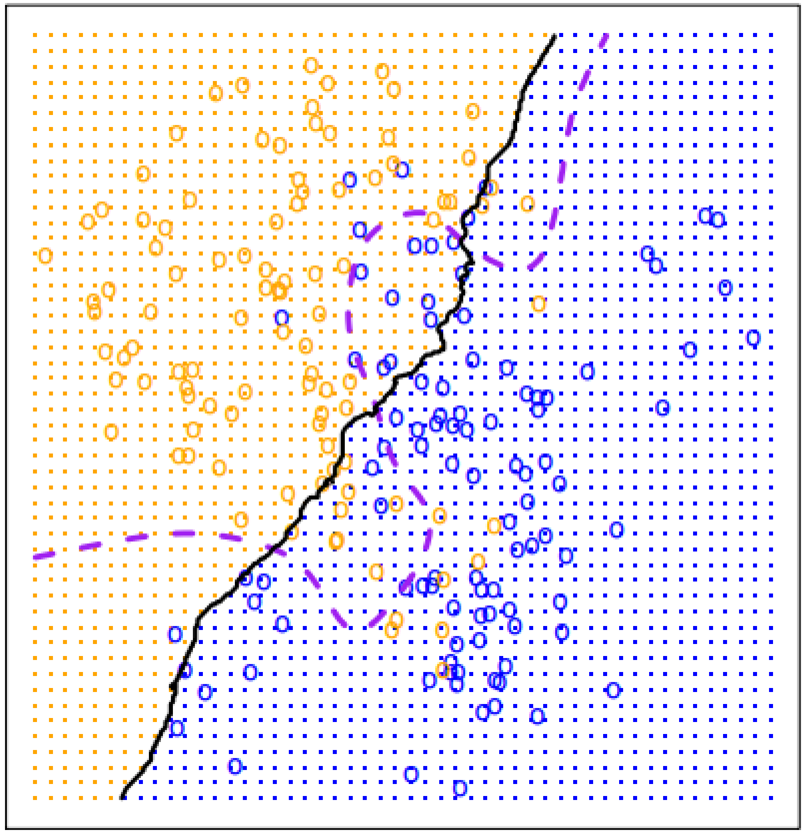
KNN: K=10



KNN: K=1

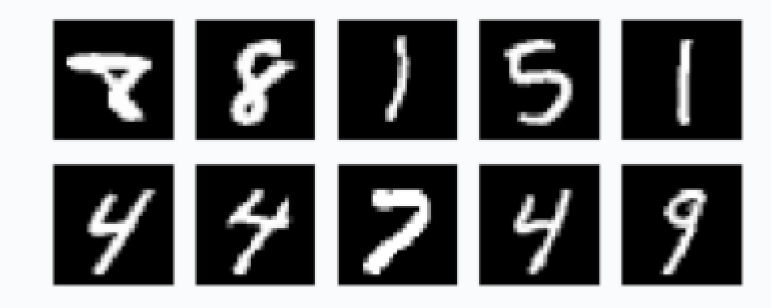


KNN: K=100



Does it work?

- MNIST: black and white image classification
  - 60k train, 10k test points
  - $d = 28 \times 28 = 784$, 10 classes (0 through 9)
  - Canonical "easy ML task"



CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
Linear Classifiers			
linear classifier (1-layer NN)	none	12.0	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	none	2.83	Kenneth Wilder, U. Chicago
K-NN with non-linear deformation (IDM)	shiftable edges	0.54	Keysers et al. IEEE PAMI 2007
K-NN with non-linear deformation (P2DHMDM)	shiftable edges	0.52	Keysers et al. IEEE PAMI 2007
2-layer NN, 300 hidden units, mean square error	none	4.7	LeCun et al. 1998
Convolutional net LeNet-4	none	1.1	LeCun et al. 1998

Some theory

- Suppose $n \to \infty$. Then $L_{1NN} \leq 2L_{Bayes}(1 - L_{Bayes})$. [Cover-Hart ’67]
  - E.g., suppose Bayes classifier makes 0 error. Then 1NN has 0 error*
    - *with infinite training data
  - Bayes classifier makes 0 error. Then 1NN has $\frac{1}{2}$ error*
  - Bayes classifier makes 0.1 error. Then 1NN has 0.18 error*
- Note that n may have to be exponentially large in d in the worst case!
  - Curse of dimensionality