Lecture 6: Classification: Nearest Neighbours Approaches

Attendance code: X6UFT2NH

Iain Styles

29 October 2019

Learning Outcomes

By the end of this lecture you should be able to:

- Understand what classication problems are
- Explain how they are similar to, and different from regression problems
- Understand and work with the MNIST dataset
- Understand and apply a simple density-based classification technique

Classification

- Regression: predict value of a continuous variable from a mixture of continuous and categorical variables.

Classification

- Regression: predict value of a continuous variable from a mixture of continuous and categorical variables.
- Classification: predict value of a *categorical* variable from a mixture of continuous and categorical variables.

Classification

- Regression: predict value of a continuous variable from a mixture of continuous and categorical variables.
- Classification: predict value of a *categorical* variable from a mixture of continuous and categorical variables.
  - Determining what type of object is present in an image.

Classification

- Regression: predict value of a continuous variable from a mixture of continuous and categorical variables.
- Classification: predict value of a *categorical* variable from a mixture of continuous and categorical variables.
  - Determining what type of object is present in an image.
  - Sorting documents into different types.

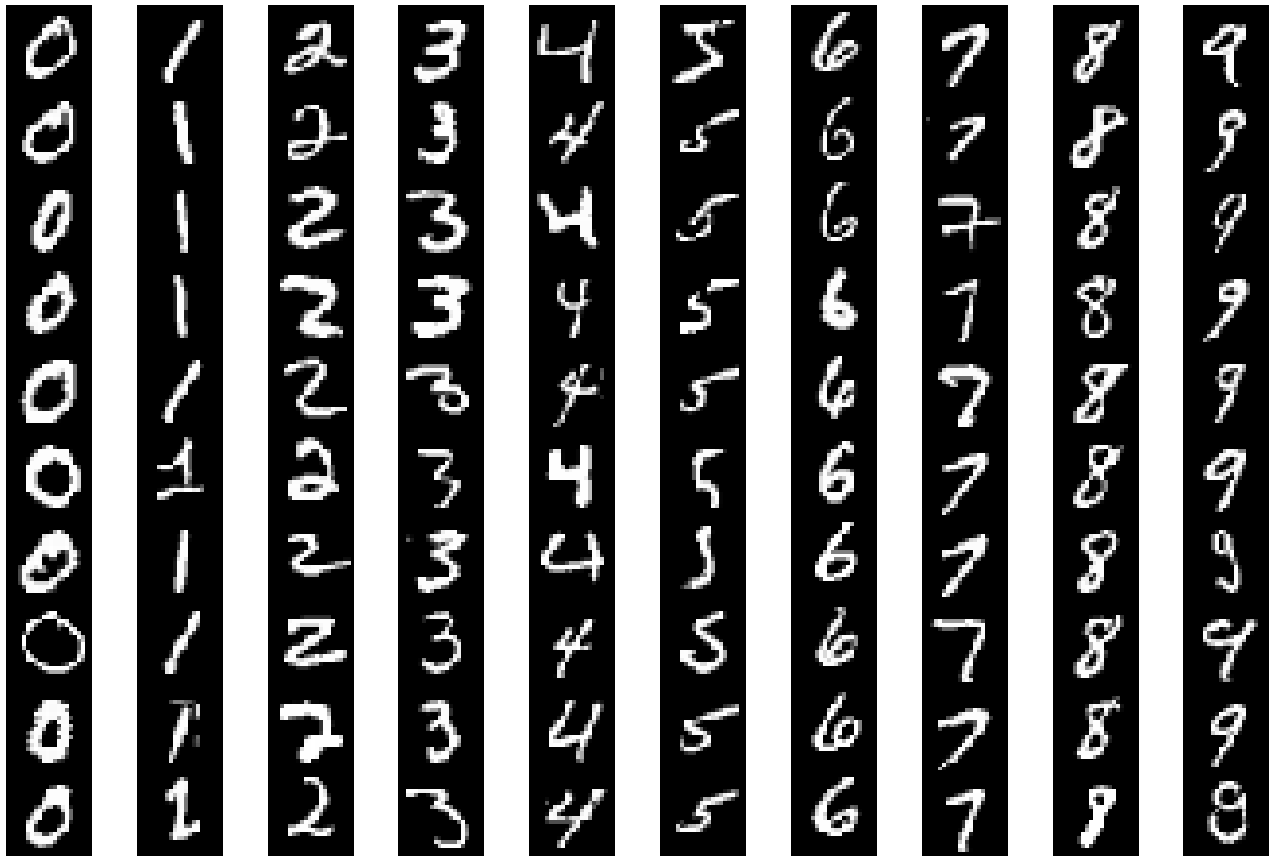
Classification

- Regression: predict value of a continuous variable from a mixture of continuous and categorical variables.
- Classification: predict value of a *categorical* variable from a mixture of continuous and categorical variables.
  - Determining what type of object is present in an image.
  - Sorting documents into different types.
  - Determining whether a set of diagnostic tests implies that a patient has a disease.
- A *supervised* learning problem: requires training samples to learn the classification rules.

Our Working Example: MNIST

- MNIST dataset of handwritten digits
- 70,000 images of characters 0–9
- All images are labelled
- 60,000 are in the *training* set
- 10,000 are in the *test* set
- All images are 28×28 pixels

MNIST Samples

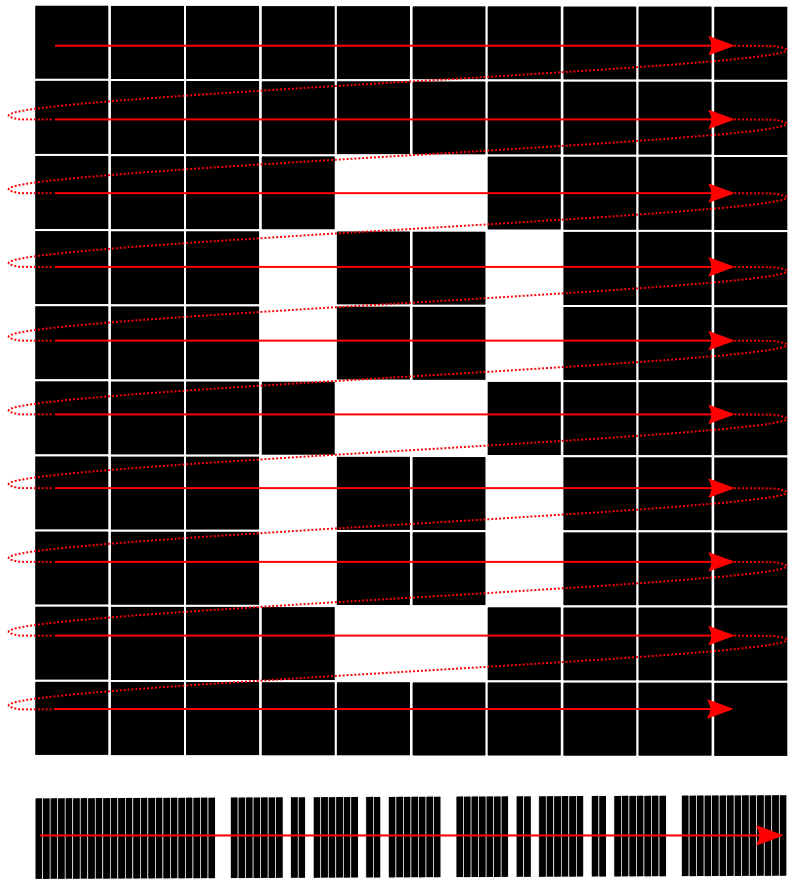


Vectorisation

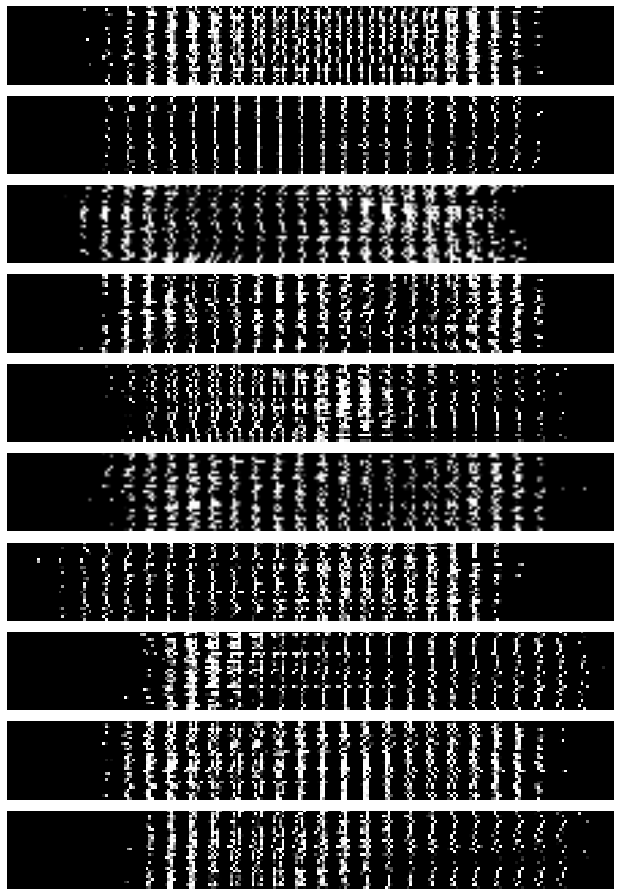
- We will study generic classification method for multovariate vectorial data
- Need to vectorise the images

Vectorisation

- We will study generic classification method for multovariate vectorial data
- Need to vectorise the images



Vectorised MNIST



A stupidly simple classification method

- Classify samples by *similarity*
- Given an unknown sample, to which sample in the training set is it most similar?

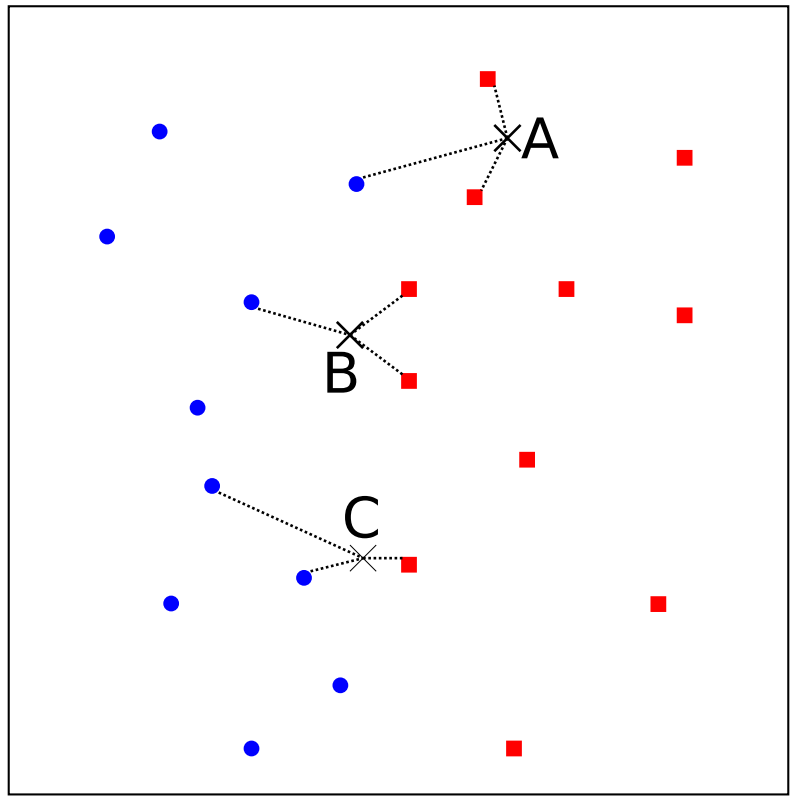
A stupidly simple classification method

- Classify samples by *similarity*
- Given an unknown sample, to which sample in the training set is it most similar?
- Nearest-neighbour classification

A stupidly simple classification method

- Classify samples by *similarity*
- Given an unknown sample, to which sample in the training set is it most similar?
- Nearest-neighbour classification
- A simple extension: take the k nearest-nighbours and assign the majority class
- k-nearest-neighbours classification

k nearest-nighbours Classification



k nearest-neighbours Classification

Data: A set of labelled training data
Data: A set of unlabelled test data
Data: Integer k
Result: For each item in the test set, returns the most common label of that items k nearest neighbours in the training set.

for *each item x in test set* **do**
  for *each item y in training set* **do**
    Compute similarity $d(x, y)$
  end
  Find the k most similar items to x in the training set.
  Compute the most common label
end

k nearest-neighbours Classification

Data: A set of labelled training data
Data: A set of unlabelled test data
Data: Integer k
Result: For each item in the test set, returns the most common label of that items k nearest neighbours in the training set.

for *each item x in test set* **do**
  for *each item y in training set* **do**
    Compute similarity $d(x, y)$
  end
  Find the k most similar items to x in the training set.
  Compute the most common label
end

- The training phase is just the data!
- Prediction is costly

*k*nn and MNIST

- Do we expect *k*nn to do well on MIST?

*k*nn and MNIST

- Do we expect *k*nn to do well on MIST?
- Vectorising the images loses much of their spatial information
- There is substantial variability between characters

*k*nn and MNIST

- Do we expect *k*nn to do well on MIST?
- Vectorising the images loses much of their spatial information
- There is substantial variability between characters
- No harm in trying...
- Need a measure of similarity: Euclidean distance
  For images vectors **x** and **y**

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{((\mathbf{x} - \mathbf{y})^{\mathrm{T}}(\mathbf{x} - \mathbf{y}))} = \sqrt{\sum_i (x_i - y_i)^2}. \qquad (1)$$

- Smaller $\rightarrow$ more similar
- Use 10,000 training samples and 1000 test samples to save time

$k = 1$ nearest-neighbours

T \ P	0	1	2	3	4	5	6	7	8	9
0	83	1	1	0	0	0	5	0	10	0
1	0	100	0	0	0	0	0	0	0	0
2	1	11	53	2	1	0	3	4	25	0
3	0	11	2	48	0	1	4	3	28	3
4	2	9	0	0	42	0	2	3	16	26
5	2	7	0	4	0	36	2	0	43	6
6	3	6	0	0	0	1	80	0	10	0
7	0	11	0	1	0	0	1	75	4	8
8	2	13	0	6	1	3	3	4	65	3
9	0	5	1	1	4	0	0	4	2	83

$k = 1$ nearest-neighbours

T \ P	0	1	2	3	4	5	6	7	8	9
0	83	1	1	0	0	0	5	0	10	0
1	0	100	0	0	0	0	0	0	0	0
2	1	11	53	2	1	0	3	4	25	0
3	0	11	2	48	0	1	4	3	28	3
4	2	9	0	0	42	0	2	3	16	26
5	2	7	0	4	0	36	2	0	43	6
6	3	6	0	0	0	1	80	0	10	0
7	0	11	0	1	0	0	1	75	4	8
8	2	13	0	6	1	3	3	4	65	3
9	0	5	1	1	4	0	0	4	2	83

- Total accuracy: 67%

$k = 3$ nearest-neighbours

T \ P	0	1	2	3	4	5	6	7	8	9
0	95	1	0	0	0	0	1	0	3	0
1	0	100	0	0	0	0	0	0	0	0
2	4	14	68	0	0	0	1	2	11	0
3	2	13	4	64	0	1	3	2	8	3
4	2	13	1	0	51	0	4	2	3	24
5	5	13	0	10	1	39	2	0	24	6
6	2	7	0	0	1	1	88	0	1	0
7	0	18	2	1	1	1	0	68	3	6
8	3	18	0	3	1	3	3	4	65	0
9	1	7	0	1	1	0	0	2	2	86

- Total accuracy: 72%

$k = 5$ nearest-neighbours

T \ P	0	1	2	3	4	5	6	7	8	9
0	97	1	0	0	0	0	1	0	1	0
1	0	100	0	0	0	0	0	0	0	0
2	3	17	69	1	0	0	2	3	5	0
3	1	19	1	60	0	0	6	3	7	3
4	2	12	1	0	50	0	5	1	4	25
5	5	9	0	5	2	51	2	0	19	7
6	2	7	0	0	1	1	89	0	0	0
7	0	18	0	0	1	1	0	73	2	5
8	3	18	1	3	0	1	4	5	65	0
9	1	9	0	0	0	0	0	1	4	85

- Total accuracy: 74%

$k = 7$ nearest-neighbours

T \ P	0	1	2	3	4	5	6	7	8	9
0	95	1	0	0	0	0	3	0	1	0
1	0	100	0	0	0	0	0	0	0	0
2	1	17	70	0	0	0	2	4	6	0
3	1	20	0	61	0	1	6	2	5	4
4	3	9	0	0	55	0	4	1	2	26
5	5	9	1	5	1	51	3	0	17	8
6	2	7	0	0	0	1	90	0	0	0
7	1	17	1	0	1	0	0	75	1	4
8	3	16	1	2	0	1	4	5	66	2
9	1	7	0	0	0	0	0	2	2	88

- Total accuracy: 75%

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric – learn it from the data

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric – learn it from the data
- Reduce the dimensionality

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric – learn it from the data
- Reduce the dimensionality – high-dimensional vector space do not behave the same way as low-dimensionality spaces

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric – learn it from the data
- Reduce the dimensionality – high-dimensional vector space do not behave the same way as low-dimensionality spaces
- Next lecture: what are the problems in high-dimensionality spaces?

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric – learn it from the data
- Reduce the dimensionality – high-dimensional vector space do not behave the same way as low-dimensionality spaces
- Next lecture: what are the problems in high-dimensionality spaces? How can we overcome them in practice?

Making it even better

- Consensus voting over k neighbours bring significant gains over single nearest neighbours
- $k = 7$: diminishing returns?
- 0, 1, 6, and 9 can be identified very accurately
- 3, 4, and 5 much more resistant
- How can we improve this?
- Change the similarity metric – learn it from the data
- Reduce the dimensionality – high-dimensional vector space do not behave the same way as low-dimensionality spaces
- Next lecture: what are the problems in high-dimensionality spaces? How can we overcome them in practice? What benefits can be realised?