ONLINE MASTERS IN **DATA SCIENCE**

DSC 255 - MACHINE LEARNING FUNDAMENTALS

IMPROVING THE PERFORMANCE OF NEAREST NEIGHBOR

SANJOY DASGUPTA, PROFESSOR

UC San Diego

COMPUTER SCIENCE & ENGINEERING

HALICIOĞLU DATA SCIENCE INSTITUTE

Nearest neighbor classification

Training images $x^{(1)}, \ldots, x^{(60000)}$
Labels $y^{(1)}, \ldots, y^{(60000)}$

To classify a new image x:

- Find its nearest neighbor amongst the $x^{(i)}$ **using Euclidean distance** in $\mathbb{R}^{784}$
- Return $y^{(i)}$

How accurate is this classifier?

Error rate of 3.09% on test set.

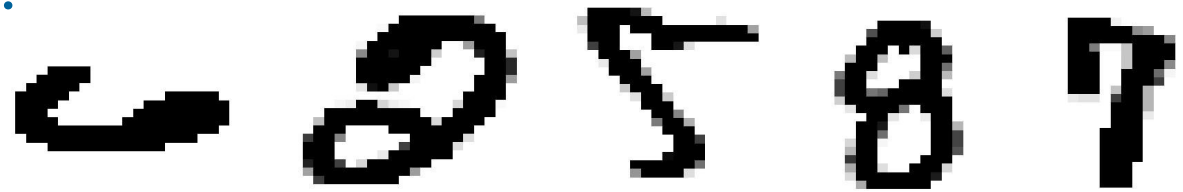
Examples of errors

Test set of 10,000 points:

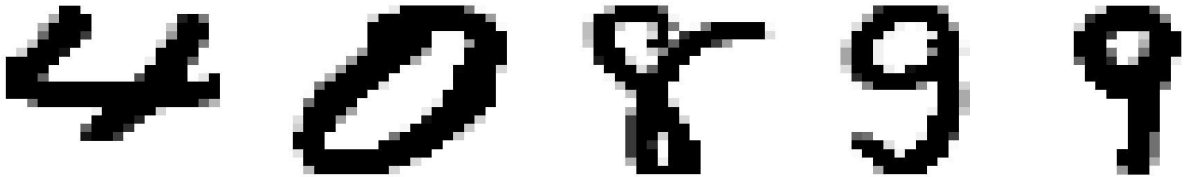
- 309 are misclassified
- Error rate 3.09%

Examples of errors:

Query



NN



Examples of errors

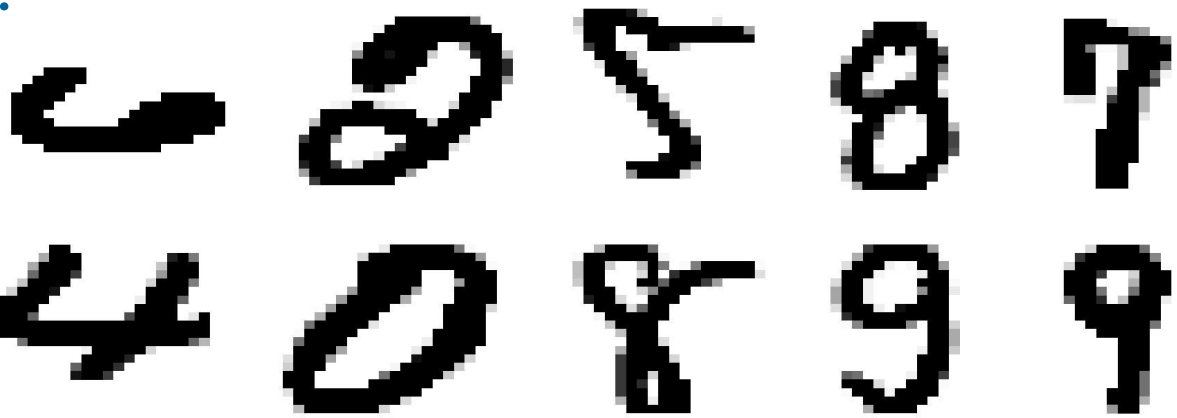
Test set of 10,000 points:

- 309 are misclassified
- Error rate 3.09%

Examples of errors:

Query

NN

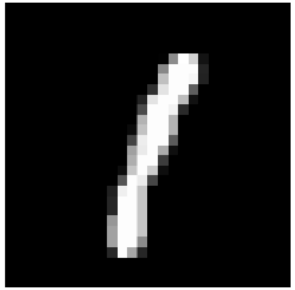


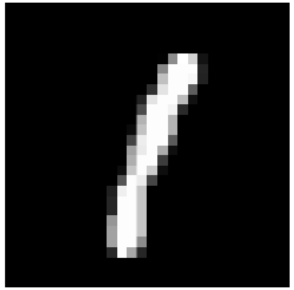
Ideas for improvement:

(1) better distance function (2) *k*-NN.

Better distance functions

The Euclidean ($\ell 2$) distance between these two images is very high!





Better distance functions

The Euclidean ($\ell 2$) distance between these two images is very high!

Much better idea: distance measures that are invariant under:

- Small translations and rotations. e.g., *tangent distance.*
- A broader family of natural deformations. e.g., *shape context.*

Better distance functions

The Euclidean ($\ell 2$) distance between these two images is very high!

Much better idea: distance measures that are invariant under:

- Small translations and rotations. e.g., *tangent distance.*
- A broader family of natural deformations. e.g., *shape context.*

Test error rates:

ℓ_2	tangent distance	shape context
3.09	1.10	0.63

Better distance functions

The Euclidean ($\ell 2$) distance between these two images is very high!

Much better idea: distance measures that are invariant under:

- Small translations and rotations. e.g., *tangent distance.*
- A broader family of natural deformations. e.g., *shape context.*

Test error rates:

ℓ_2	tangent distance	shape context
3.09	1.10	0.63

More generally: better representations for nearest neighbor.

Another improvement: *K*-nearest neighbor classification

Classify a point using the labels of its k-nearest neighbors among the training points.

Another improvement: *K*-nearest neighbor classification

Classify a point using the labels of its k-nearest neighbors among the training points.

MNIST:

K	1	3	5	7	9	11
Test error (%)	3.09	2.94	3.13	3.10	3.43	3.34

Another improvement: *K*-nearest neighbor classification

Classify a point using the labels of its k-nearest neighbors among the training points.

MNIST:

K	1	3	5	7	9	11
Test error (%)	3.09	2.94	3.13	3.10	3.43	3.34

Problem: In real life, there's no test set. How to decide which k is best?