InFoMM CDT Scientific Computing

Image Classification Task

October 16, 2017

Image classification

- Suppose we have a collection of images, each of which belongs to a (unknown) class

Image classification

- Suppose we have a collection of images, each of which belongs to a (unknown) class

For example:

- Images of faces (perhaps in different orientations, expressions, lighting
  $\Longrightarrow$ the classes are the people
- Diagnosis from medical images: the classes might be positive/negative
- Military applications: in radar for example - who exactly does that aircraft belong to?
- Document classification from handwriting style.

The two main types of classification problem

- **Do we have prior information?**

The two main types of classification problem

- **Do we have prior information?**
- In other words, do we have examples of images from each of the classes?

The two main types of classification problem

- **Do we have prior information?**
- In other words, do we have examples of images from each of the classes?
- This means we need not just images but *class labels*

The two main types of classification problem

- **Do we have prior information?**
- In other words, do we have examples of images from each of the classes?
- This means we need not just images but *class labels*
- **Supervised classification:** Use prior information to learn about the classes, then use the information to match unknown data

The two main types of classification problem

- **Do we have prior information?**
- In other words, do we have examples of images from each of the classes?
- This means we need not just images but *class labels*
- **Supervised classification:** Use prior information to learn about the classes, then use the information to match unknown data
- **Unsupervised classification:** Discover clusters in the data without using any prior information

The two main types of classification problem

- **Do we have prior information?**
- In other words, do we have examples of images from each of the classes?
- This means we need not just images but *class labels*
- **Supervised classification:** Use prior information to learn about the classes, then use the information to match unknown data
- **Unsupervised classification:** Discover clusters in the data without using any prior information
- Our project will focus on **supervised classification**.

Test data sets

- Suppose we want to compare different approaches to classification
- On real problems, the answer is unknown (that's why we're doing it!)
  $\Longrightarrow$ even it we do well, we might not know...

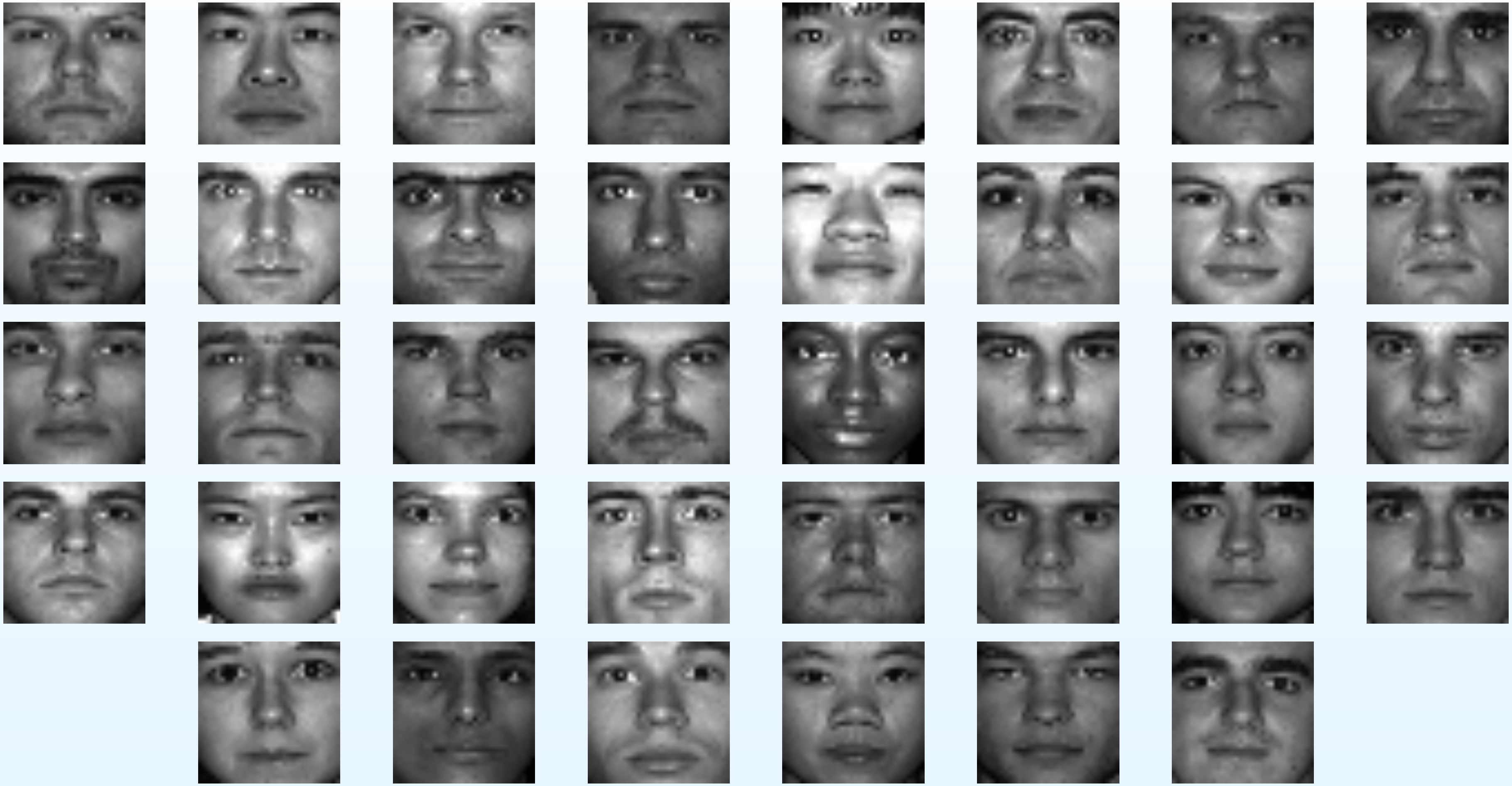
Test data sets

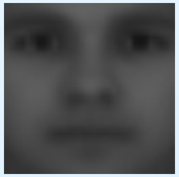
- Suppose we want to compare different approaches to classification
- On real problems, the answer is unknown (that's why we're doing it!)
  $\Longrightarrow$ even it we do well, we might not know...
- So instead we use test data sets where
  - the classes of the images being matched are known...
  - ...but we're not allowed to use them...
  - ...we can only use the information to see how well we did afterwards.

Test data sets

- Suppose we want to compare different approaches to classification
- On real problems, the answer is unknown (that's why we're doing it!)
  $\Longrightarrow$ even it we do well, we might not know...
- So instead we use test data sets where
  - the classes of the images being matched are known...
  - ...but we're not allowed to use them...
  - ...we can only use the information to see how well we did afterwards.
- $\Longrightarrow$ **two sets: training set and test set**

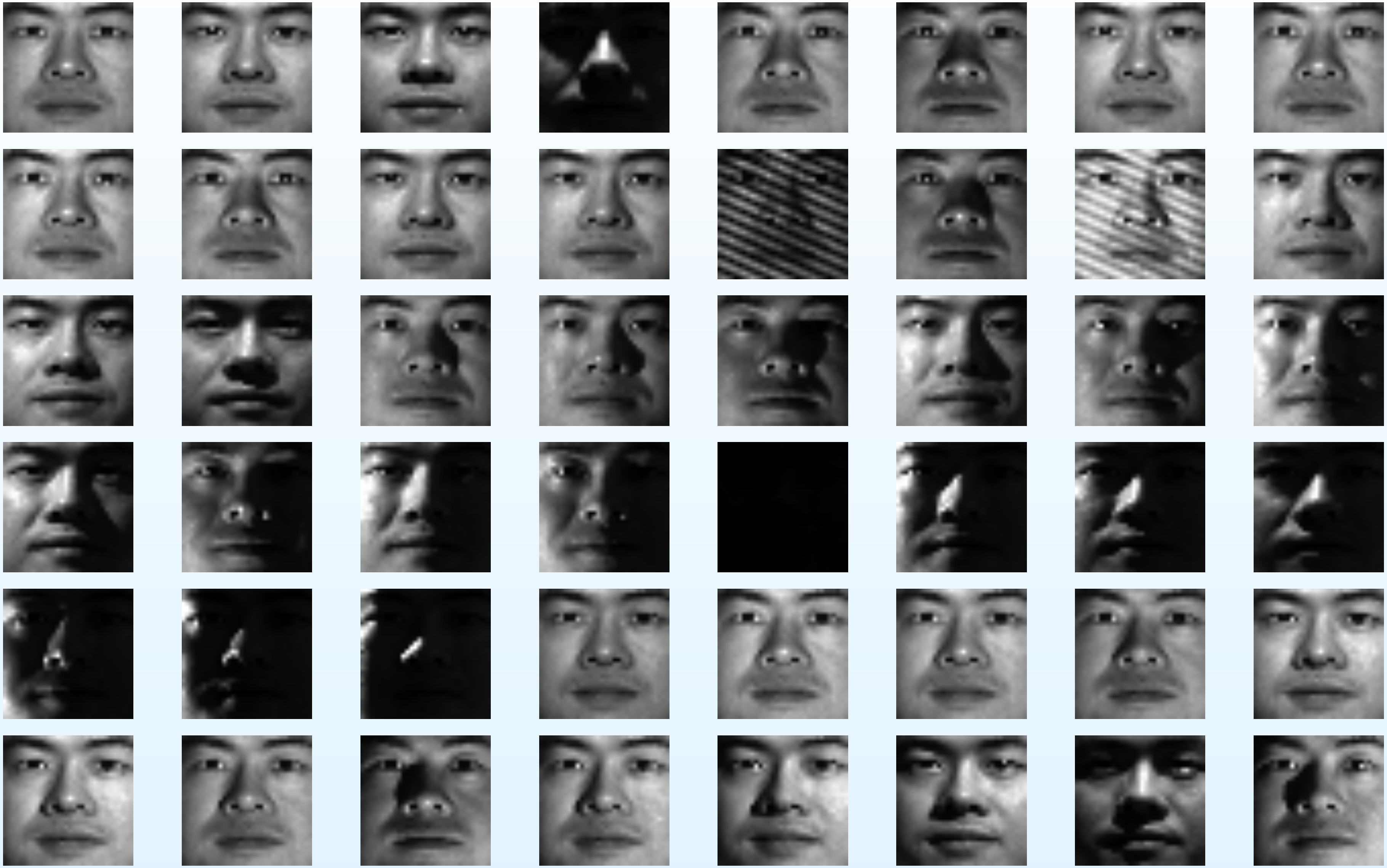
The Yale face database



...and the mean face: 

The Yale face database

The Yale face database



The Yale face database

k-nearest neighbours algorithm

	Training feature vectors	x_i	$i = 1, \dots, m$
Input:	Training labels	g_i	$i = 1, \dots, m$
	Test feature vectors	y_i	$j = 1, \dots, n$

k-nearest neighbours algorithm

Input:	Training feature vectors	x_i	$i = 1, \ldots, m$
	Training labels	g_i	$i = 1, \ldots, m$
	Test feature vectors	y_i	$j = 1, \ldots, n$

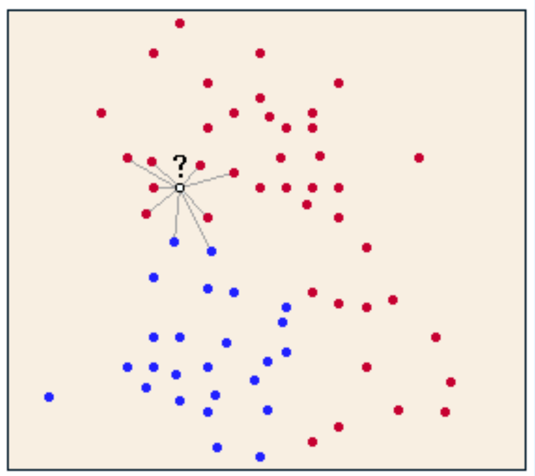
For each y_j:

k-nearest neighbours algorithm

	Training feature vectors	x_i	$i = 1, \ldots, m$
Input:	Training labels	g_i	$i = 1, \ldots, m$
	Test feature vectors	y_i	$j = 1, \ldots, n$

For each y_j:

- Find the k feature vectors which are closest to y_j.

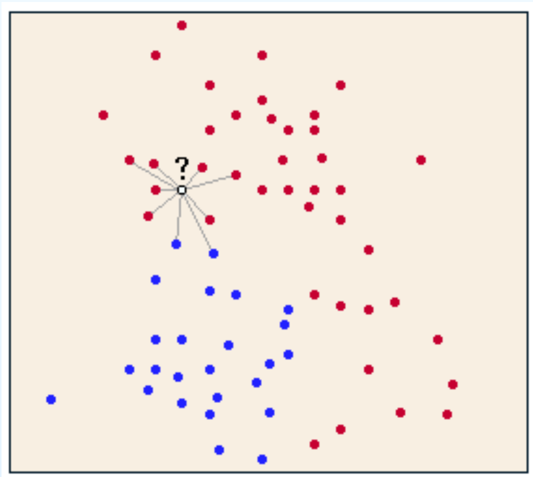


k-nearest neighbours algorithm

	Training feature vectors	x_i	$i = 1, \ldots, m$
Input:	Training labels	g_i	$i = 1, \ldots, m$
	Test feature vectors	y_i	$j = 1, \ldots, n$

For each y_j:

- Find the k feature vectors which are closest to y_j.



- Assign the most frequent label among the selected x_i.

What is a digital image?

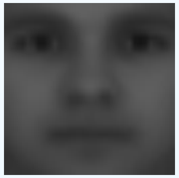
82	93	108	116	131	140	142	139	162	172	171	169	174	155	152	151	159	166	171	182	173	181	179	162	166	156	144	122	113	97	92	71
81	96	106	127	137	153	153	166	160	188	193	194	200	181	198	209	131	200	220	222	220	205	185	175	170	145	141	133	116	110	94	70
72	90	113	119	123	137	145	155	163	163	186	187	194	197	181	190	197	193	208	172	204	196	174	161	139	118	113	103	75	67	93	81
72	97	60	64	73	78	84	85	84	102	105	127	145	163	177	189	190	172	172	138	125	104	84	74	76	69	95	135	127	96	76	46
40	68	103	123	145	141	125	85	70	59	71	85	110	136	154	162	170	158	139	121	91	77	68	84	131	146	137	138	127	123	116	87
93	120	126	126	128	134	134	131	108	93	80	77	92	115	143	170	164	144	121	103	87	87	97	82	82	91	81	85	109	112	111	109
119	109	99	75	68	85	64	56	56	82	91	92	84	90	120	167	156	130	97	91	93	86	84	72	30	48	28	57	59	79	103	109
129	111	76	57	42	109	20	18	51	200	75	138	89	97	92	140	146	111	84	92	158	98	97	189	23	23	33	120	33	54	90	112
135	105	63	59	80	91	95	103	103	105	112	114	115	96	87	128	133	107	82	99	145	123	107	96	93	87	83	92	75	59	65	100
121	123	117	107	112	123	123	127	137	121	112	122	107	74	74	101	108	90	78	91	110	132	114	112	130	120	121	116	111	113	108	111
123	126	124	125	132	142	149	158	151	134	128	119	85	67	82	103	101	87	75	87	98	124	140	143	161	170	163	144	138	127	117	113
124	130	137	132	138	149	160	169	159	153	148	117	75	62	93	108	103	85	68	77	90	128	143	162	165	171	163	159	150	139	137	126
130	127	134	134	138	152	152	165	158	143	136	95	57	87	105	111	108	95	69	63	87	112	137	154	162	175	177	159	150	141	143	127
126	124	135	132	140	150	145	151	133	146	133	88	71	91	124	142	138	114	97	75	71	112	143	152	146	165	167	153	143	141	144	124
116	121	122	122	140	149	149	138	147	160	139	64	98	104	155	183	191	126	99	105	70	116	153	148	156	151	147	153	142	138	128	122
107	105	118	128	140	147	145	144	161	168	143	99	113	83	135	155	145	99	93	126	101	109	167	176	166	151	139	132	141	132	126	121
94	105	112	117	123	147	150	154	178	187	131	106	121	65	66	110	96	79	67	92	110	122	155	190	171	154	152	149	134	130	117	111
94	96	108	122	125	150	163	152	191	178	145	95	96	96	129	112	115	127	91	98	75	144	163	181	185	157	153	146	143	118	114	107
90	91	118	137	133	158	151	184	178	150	140	103	78	88	133	135	141	118	84	88	102	132	146	165	185	181	159	155	150	117	112	100
86	104	115	134	152	160	170	182	149	128	109	101	75	80	123	125	116	119	87	93	93	113	130	138	164	179	161	147	137	104	110	91
73	101	123	111	131	139	161	144	92	107	93	91	83	91	125	139	123	134	97	87	81	89	111	109	112	163	172	143	145	127	107	95
89	97	102	123	121	133	127	101	103	103	88	87	108	123	121	113	121	139	126	110	91	88	95	105	96	111	134	146	142	126	113	99
76	87	109	125	136	109	95	101	94	127	109	118	116	122	113	114	118	124	112	111	111	113	116	107	107	91	123	137	134	131	109	91
80	72	103	121	108	96	113	114	130	118	109	98	84	83	84	78	81	85	87	83	86	111	115	125	124	101	103	92	140	109	92	89
68	82	93	90	113	100	124	125	112	83	69	52	64	67	74	81	70	78	68	59	62	63	68	72	121	115	101	103	132	103	94	92
54	80	90	104	97	99	118	136	117	70	75	80	104	112	111	111	112	109	107	103	89	81	73	100	143	134	113	122	108	85	95	75
74	84	79	85	107	106	109	110	117	92	93	108	127	127	127	133	125	134	137	133	118	110	100	102	132	133	120	128	86	82	89	80
59	65	68	85	114	95	108	119	115	115	118	124	119	115	107	110	116	121	131	133	137	130	123	121	113	123	113	100	89	92	76	67
43	52	60	72	79	98	101	107	112	116	118	125	107	112	109	115	114	124	134	134	137	123	101	112	117	94	97	106	73	67	61	57
36	43	48	52	72	71	86	95	106	105	125	137	144	149	127	124	122	112	126	152	140	132	110	109	100	82	91	68	69	60	55	35
21	23	32	41	46	55	77	81	82	69	113	134	130	116	104	122	124	104	106	139	130	118	115	102	76	76	55	57	50	51	55	33
12	17	26	30	33	58	45	63	66	86	99	125	102	125	131	124	139	129	133	125	89	104	87	69	70	50	59	36	39	37	34	22

A simple feature vector: intensity

- Vectorize each training image x_i to give a column of X, a $p \times m$ matrix where p is the number of pixels

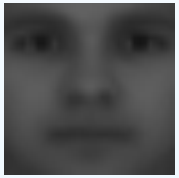
A simple feature vector: intensity

- Vectorize each training image x_i to give a column of X, a $p \times m$ matrix where p is the number of pixels
- Calculate the mean image: `Xm = sum(X,2)/m`



A simple feature vector: intensity

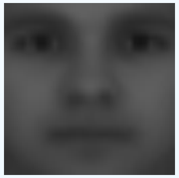
- Vectorize each training image x_i to give a column of X, a $p \times m$ matrix where p is the number of pixels
- Calculate the mean image: `Xm = sum(X,2)/m`



- Subtract the mean from each image: `Xc = X - repmat(Xm,[1 m])`

A simple feature vector: intensity

- Vectorize each training image x_i to give a column of X, a $p \times m$ matrix where p is the number of pixels
- Calculate the mean image: `Xm = sum(X,2)/m`



- Subtract the mean from each image: `Xc = X - repmat(Xm,[1 m])`
- Do the same for the test images: `Yc = Y - repmat(Xm,[1 n])`

Principal component analysis features

- Consider the $p \times p$ **covariance matrix** XX^T
- Take its eigendecomposition `[V D] = eig(X*X')`

Principal component analysis features

- Consider the $p \times p$ **covariance matrix** XX^T
- Take its eigendecomposition `[V D] = eig(X*X')`
- Let $V = [V_1 \; V_2]$ where the columns of V_1 correspond to the largest (in magnitude) eigenvalues $d_1, d_2, \ldots, d_k$, $k < m$

Principal component analysis features

- Consider the $p \times p$ **covariance matrix** XX^T
- Take its eigendecomposition `[V D] = eig(X*X')`
- Let $V = [V_1 \ V_2]$ where the columns of V_1 correspond to the largest (in magnitude) eigenvalues $d_1, d_2, \ldots, d_k, k < m$
- Project into the space spanned by V_1:

```
Xf=V'*X; Xf=flipud(Xf); Xf=Xf(1:k,:)
```

- Do the same with Y: project into V_1
- We now have smaller feature vectors!

Principal component analysis features

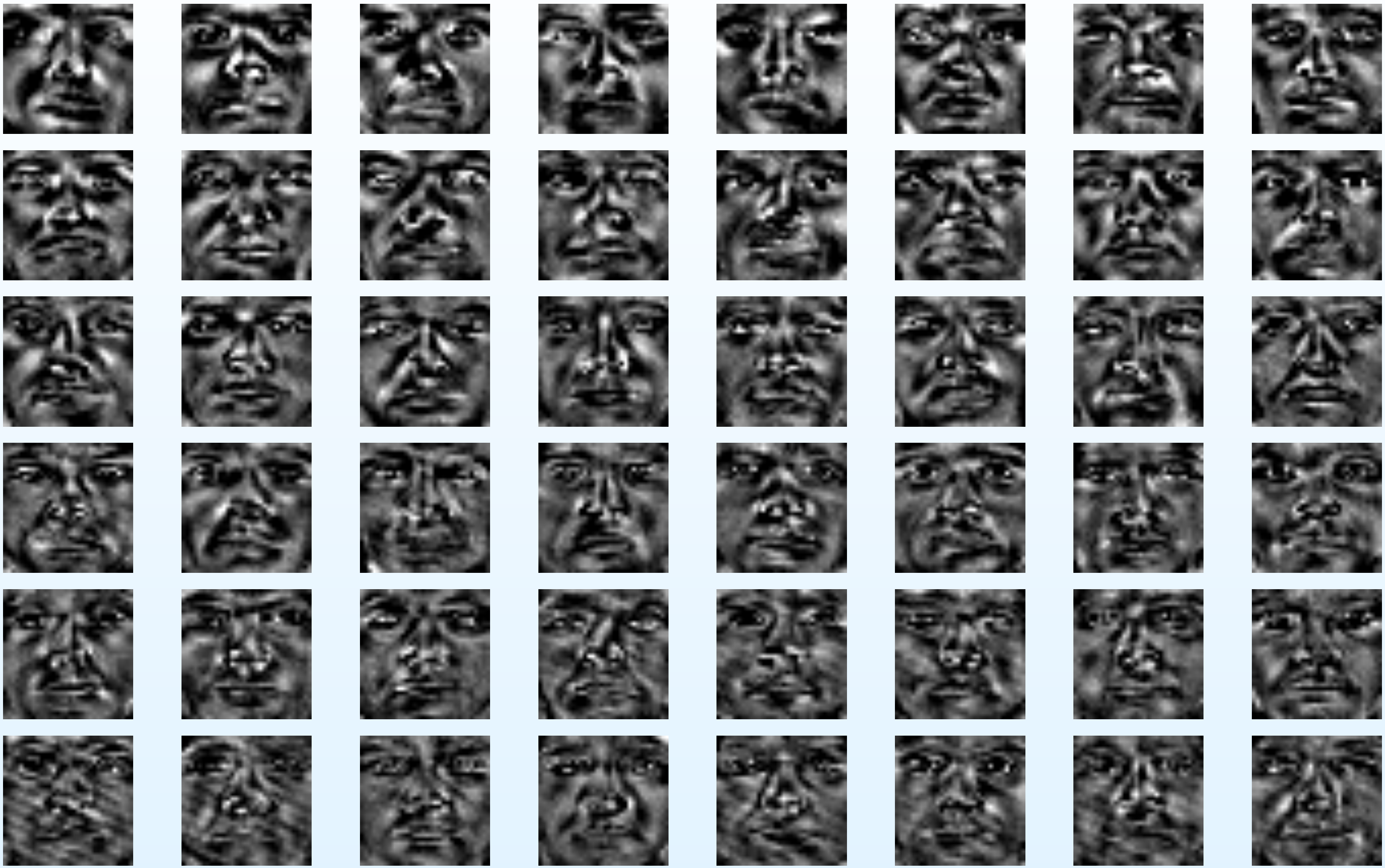
- Consider the $p \times p$ **covariance matrix** XX^T
- Take its eigendecomposition `[V D] = eig(X*X')`
- Let $V = [V_1 \; V_2]$ where the columns of V_1 correspond to the largest (in magnitude) eigenvalues $d_1, d_2, \ldots, d_k$, $k < m$
- Project into the space spanned by V_1:

```
Xf=V'*X; Xf=flipud(Xf); Xf=Xf(1:k,:)
```

- Do the same with Y: project into V_1
- We now have smaller feature vectors!
- If $m < p$, we can first do `[U D] = eig(X'*X)`, and then obtain the eigenvectors of XX^T as `V=X*U`.

Eigenfaces

Eigenfaces



Improvements to eigenfaces

- **Whitening:** Normalize the features:
  In place of `Xf=V'*X`, do `Xf=D.^(-1/2)*V'*X`.

Improvements to eigenfaces

- **Whitening:** Normalize the features:
  In place of `Xf=V'*X`, do `Xf=D.^(-1/2)*V'*X`.

- **Removal of first few components:**
  The first few eigenvectors capture lighting effects more than distinguishing people.

Some example matchings



-2.2057	-1.8430	1.2718	-0.3012	-3.9377	1.5653	4.4381	2.5084	0.7554	0.0941	5.2085
-1.5632	-5.5112	4.9246	2.6663	-2.8053	5.2122	7.8306	-0.3192	1.5501	0.0609	5.7439
-0.7094	3.1290	1.5869	5.2848	0.6720	3.1660	3.8727	2.1090	2.9190	0.2547	4.1439
-7.6463	0.3238	-1.4605	5.8808	-6.3703	2.2854	1.4323	4.8371	2.2619	-0.3084	3.6800

Some example matchings

-0.2481	-0.1787	0.9437	0.4573	-0.2593	0.7267	0.0941	-0.0009	0.3398	-0.1720	0.1519
-0.2797	-0.2745	0.9645	0.4286	-0.1054	0.7057	0.0398	-0.1230	0.3175	-0.4393	0.0914
-0.3654	-0.1372	0.8560	0.4541	-0.4664	0.4134	0.0388	-0.0432	0.3673	-0.3042	0.1804
-0.4540	-0.1347	1.2253	0.4417	-0.0277	0.7711	0.0818	-0.0573	0.3836	-0.3561	0.0990

-2.2057	-1.8430	1.2718	-0.3012	-3.9377	1.5653	4.4381	2.5084	0.7554	0.0941	5.2085
-1.5632	-5.5112	4.9246	2.6663	-2.8053	5.2122	7.8306	-0.3192	1.5501	0.0609	5.7439
-0.7094	3.1290	1.5869	5.2848	0.6720	3.1660	3.8727	2.1090	2.9190	0.2547	4.1439
-7.6463	0.3238	-1.4605	5.8808	-6.3703	2.2854	1.4323	4.8371	2.2619	-0.3084	3.6800

Visualizing the clustering

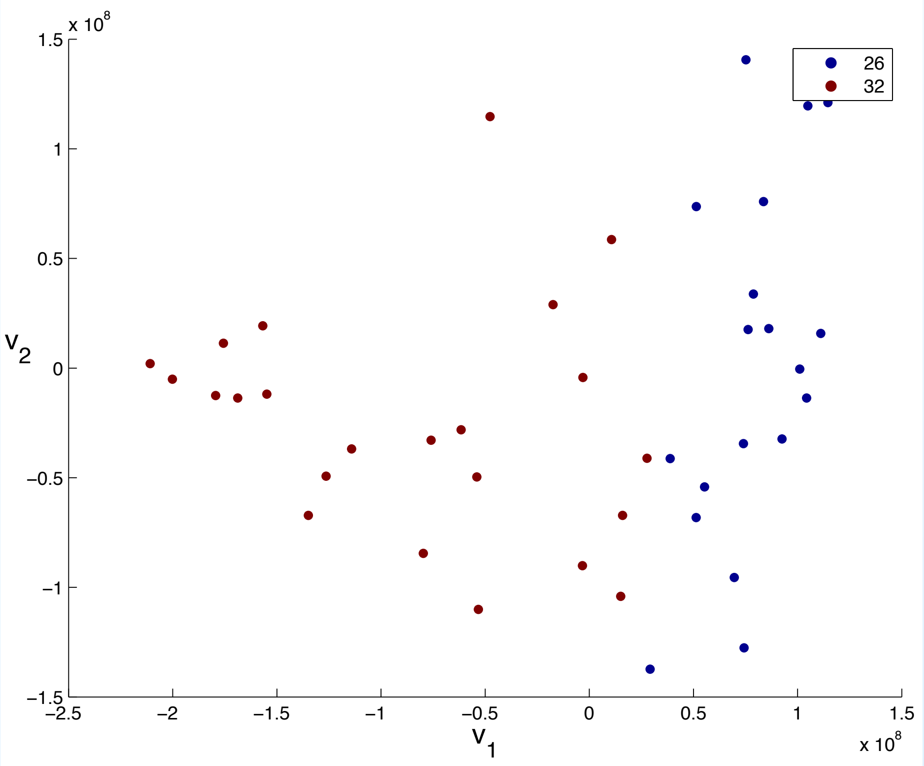
- Project onto the first two (or three?) principal components (eigenvectors) of the feature set

Visualizing the clustering

- Project onto the first two (or three?) principal components (eigenvectors) of the feature set
- This means precisely taking the first two coefficients in the PCA feature vector

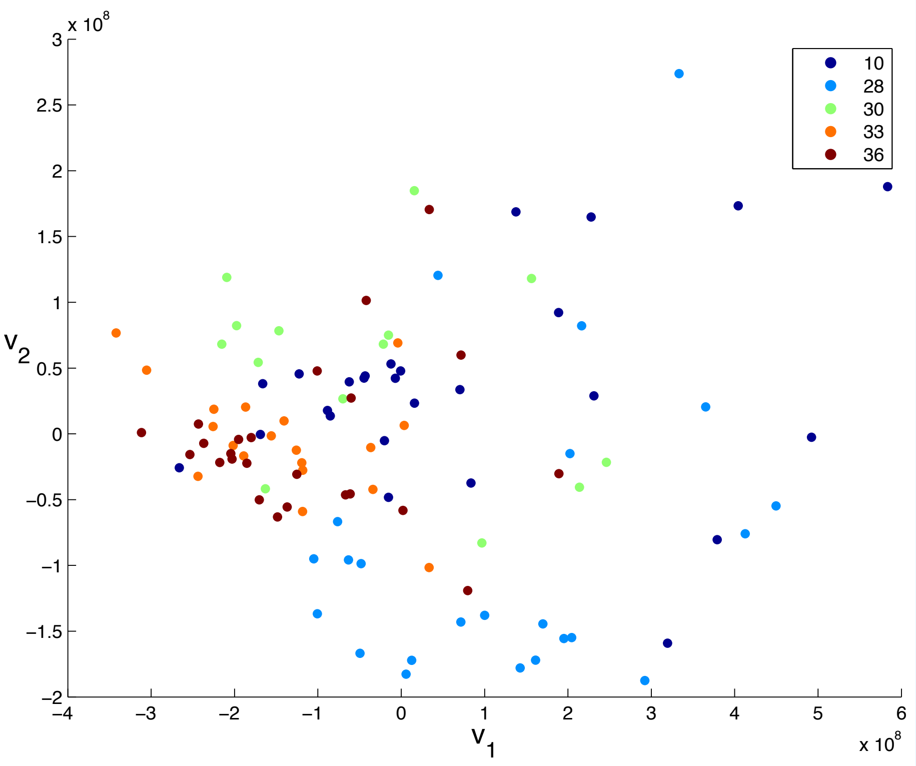
Visualizing the clustering

- Project onto the first two (or three?) principal components (eigenvectors) of the feature set
- This means precisely taking the first two coefficients in the PCA feature vector



Visualizing the clustering

- Project onto the first two (or three?) principal components (eigenvectors) of the feature set
- This means precisely taking the first two coefficients in the PCA feature vector



Your task

- **Write a suite of code in MATLAB for testing supervised classification.**

Your task

- **Write a suite of code in MATLAB for testing supervised classification.**

Use...

- the MNIST database
- the k-nearest neighbour classifier

Your task

- **Write a suite of code in MATLAB for testing supervised classification.**

Use...

- the MNIST database
- the k-nearest neighbour classifier

Provide choices for...

- which digits to test
- the size of training/test sets
- the features to use (e.g. intensity, PCA...)

Possible extensions

- Visualization of clustering using PCA projection

Possible extensions

- Visualization of clustering using PCA projection
- Exploration of other (improved?) features

Possible extensions

- Visualization of clustering using PCA projection
- Exploration of other (improved?) features
- Use of classifiers other than k-nearest neighbours

Possible extensions

- Visualization of clustering using PCA projection
- Exploration of other (improved?) features
- Use of classifiers other than k-nearest neighbours
- Inclusion of unsupervised classification

Possible extensions

- Visualization of clustering using PCA projection
- Exploration of other (improved?) features
- Use of classifiers other than k-nearest neighbours
- Inclusion of unsupervised classification
- Extension to other image databases