F20DL and F21DL: Part 2 Machine Learning
Lecture 3: Unsupervised Learning: clustering

Katya Komendantskaya

Our schedule and where we are

CW	Lectures	Week
Test 1	Bayesian Learning	Week 6
Test 2	Clustering	Week 7
CW2	“Reading week”	Week 8
Test 3	Decision trees and Regression	Week 9
Test 4	Neural Nets	Week 10
CW3	“Revision week”	Weeks 11

Our schedule and where we are

CW	Lectures	Week
Test 1	Bayesian Learning	Week 6
Test 2	Clustering	Week 7
CW2	"Reading week"	Week 8
Test 3	Decision trees and Regression	Week 9
Test 4	Neural Nets	Week 10
CW3	"Revision week"	Weeks 11

Note: no Thursday lecture on Week 9, instead, a Thursday lecture on Week 8. There will be labs on Week 8, just in case you need help with Test 2 or CW2.

Note: test exercises = lab exercises.

Last week

… we discussed

- Bayesian Learning, Bayes Nets
- Learning was defined as Knowledge revision (more precisely, computation of posterior probabilities)

Last week

... we discussed

- Bayesian Learning, Bayes Nets
- Learning was defined as Knowledge revision (more precisely, computation of posterior probabilities)

Today:

- A related kind of learning – Unsupervised Learning, or clustering.
- Learning is about finding a good model: learning as search

Basic intuition: supervised learning

	X				X		
			X				
	X				X		
		X	X	X			

	X				X		
			X				
X						X	
	X				X		
		X	X	X			

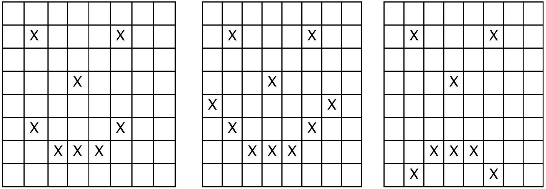
	X				X		
			X				
		X	X	X			
	X				X		

Basic intuition: supervised learning

Picture	Cell 33	Cell 42	Cell 48	Cell 58	Face expression
P1	White	Black	White	White	Happy
P2	Black	Black	White	White	Happy
P3	White	White	White	Black	Sad
P4	White	White	Black	White	Sad
P5	Black	White	Black	Black	Happy
P6	White	White	Black	Black	Sad
P7	Black	White	White	Black	Sad
P8	Black	White	Black	Black	Sad
P9	White	Black	Black	Black	Sad
P10	White	Black	White	Black	Sad

Why are we talking of "supervision here?"

Basic intuition: unsupervised learning



There was no-one to mark pictures as happy or sad for us!

Basic intuition: unsupervised learning

It will look like this:

Picture	Cell 33	Cell 42	Cell 48	Cell 58	
P1	White	Black	White	White	
P2	Black	Black	White	White	
P3	White	White	White	Black	
P4	White	White	Black	White	
P5	Black	White	Black	Black	
P6	White	White	Black	Black	
P7	Black	White	White	Black	
P8	Black	White	Black	Black	
P9	White	Black	Black	Black	
P10	White	Black	White	Black	

Learning as search

- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.

Learning as search

- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.
- Learning is search through the space of possible representations looking for the representation or representations that best fits the data, given the bias.

Learning as search

- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.
- Learning is search through the space of possible representations looking for the representation or representations that best fits the data, given the bias.
- These search spaces are typically prohibitively large for systematic search. E.g., use gradient descent.

Learning as search

- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.
- Learning is search through the space of possible representations looking for the representation or representations that best fits the data, given the bias.
- These search spaces are typically prohibitively large for systematic search. E.g., use gradient descent.
- A learning algorithm is made of a search space, an evaluation function, and a search method.

Characterizations of Learning

One of the three possible:

- Find the best representation given the data.

Characterizations of Learning

One of the three possible:

- Find the best representation given the data.
- Delineate the class of consistent representations given the data.

Characterizations of Learning

One of the three possible:

- Find the best representation given the data.
- Delineate the class of consistent representations given the data.
- Find a probability distribution of the representations given the data.

Clustering / Unsupervised Learning

- The target (class) features are not given in the training examples.

Clustering / Unsupervised Learning

- The target (class) features are not given in the training examples.
- The aim is to construct a natural classification that can be used to predict features of the data.

Clustering / Unsupervised Learning

- The target (class) features are not given in the training examples.
- The aim is to construct a natural classification that can be used to predict features of the data.
- The examples are partitioned into clusters or classes. Each class predicts feature values for the examples in the class.

Clustering / Unsupervised Learning

- The target (class) features are not given in the training examples.
- The aim is to construct a natural classification that can be used to predict features of the data.
- The examples are partitioned into clusters or classes. Each class predicts feature values for the examples in the class.
  - In hard clustering each example is placed definitively in a class.
  - In soft clustering each example has a probability distribution over its class.
- Each cluster has a prediction error on the examples. The best clustering is the one that minimizes the error.

Expectation Maximisation

... common name for soft and hard clustering algorithms that follow the scheme:

- Start with random assignment of examples to classes
- E Classify the data using the current theory (generates expected classification for each example)
- M Generate the best theory using the current classification of the data (generates the most likely theory given the classified data)
- Repeat steps E and M until the algorithm converges to the “best” class assignment

k-means algorithm

The k-means algorithm is an EM algorithm used for hard clustering.

Inputs:

- training examples
- the number of classes, k

Outputs:

- a prediction of a value for each feature for each class
- an assignment of examples to classes

k-means algorithm formalized

- E is the set of all examples
- the input features are $X_1, \dots, X_n$
- $val(e, X_j)$ is the value of feature X_j for example e.
- there is a class for each integer $i \in \{1, \dots, k\}$.

k-means algorithm formalized

- E is the set of all examples
- the input features are $X_1, \ldots, X_n$
- $val(e, X_j)$ is the value of feature X_j for example e.
- there is a class for each integer $i \in \{1, \ldots, k\}$.

The k-means algorithm outputs

- a function $class : E \to \{1, \ldots, k\}$.
  $class(e) = i$ means e is in class i.

k-means algorithm formalized

- E is the set of all examples
- the input features are $X_1, \ldots, X_n$
- $val(e, X_j)$ is the value of feature X_j for example e.
- there is a class for each integer $i \in \{1, \ldots, k\}$.

The k-means algorithm outputs

- a function $class : E \rightarrow \{1, \ldots, k\}$.
  $class(e) = i$ means e is in class i.
- a $pval$ function where $pval(i, X_j)$ is the prediction for each example in class i for feature X_j.

k-means algorithm formalized

- E is the set of all examples
- the input features are $X_1, \ldots, X_n$
- $val(e, X_j)$ is the value of feature X_j for example e.
- there is a class for each integer $i \in \{1, \ldots, k\}$.

The k-means algorithm outputs

- a function $class : E \rightarrow \{1, \ldots, k\}$.
  $class(e) = i$ means e is in class i.
- a $pval$ function where $pval(i, X_j)$ is the prediction for each example in class i for feature X_j.

The sum-of-squares error for $class$ and $pval$ is

$$\sum_{e \in E} \sum_{j=1}^{n} \left(pval(class(e), X_j) - val(e, X_j) \right)^2 .$$

Aim: find $class$ and $pval$ that minimize sum-of-squares error.

Minimizing the error

The sum-of-squares error for *class* and *pval* is

$$\sum_{e \in E} \sum_{j=1}^{n} (pval(class(e), X_j) - val(e, X_j))^2 \, .$$

- Given *class*, the *pval* that minimizes the sum-of-squares error is the mean value for that class.
- Given *pval*, each example can be assigned to the class that minimizes the error for that example.

Minimizing the error

The sum-of-squares error for *class* and *pval* is

$$\sum_{e \in E} \sum_{j=1}^{n} \left(pval(class(e), X_j) - val(e, X_j) \right)^2 .$$

- Given *class*, the *pval* that minimizes the sum-of-squares error is the mean value for that class.
- Given *pval*, each example can be assigned to the class that minimizes the error for that example.

Another name for the formula – Eucledian distance metric

This is why, if you have n examples, each given by m features, you will effectively be clustering n points in m-dimensional space.

k-means algorithm

Initially, randomly assign the examples to the classes.
Repeat the following two steps:

M For each class i and feature X_j,

$$pval(i, X_j) = \frac{\sum_{e:class(e)=i} val(e, X_j)}{|\{e : class(e) = i\}|},$$

(Another name for *pval* – centroid)

k-means algorithm

Initially, randomly assign the examples to the classes.

Repeat the following two steps:

M For each class i and feature X_j,

$$pval(i, X_j) = \frac{\sum_{e:class(e)=i} val(e, X_j)}{|\{e : class(e) = i\}|},$$

(Another name for *pval* – centroid)

E For each example e, assign e to the class i that minimizes

$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 .$$

until the second step does not change the assignment of any example.

Example Data

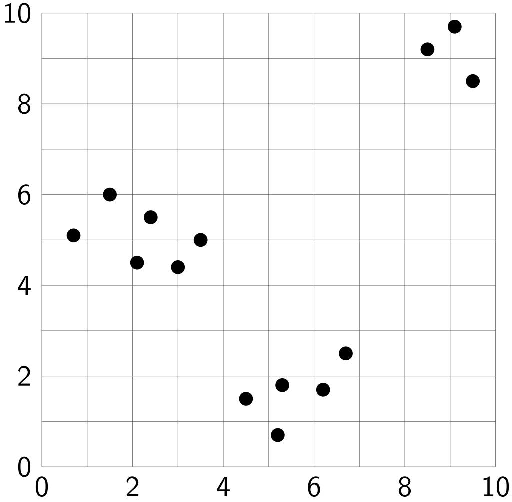
Example

Take the following objects (each represented by two features):
(0,7; 5,1), (1,5; 6), (2,1; 4,5), (2,4; 5,5), (3; 4,4),
(3,5; 5), (4,5; 1,5), (5,2; 0,7), (5,3; 1,8), (6,2; 1,7), (6,7; 2,5),
(8,5; 9,2), (9,1; 9,7), (9,5; 8,5).

As a data set:

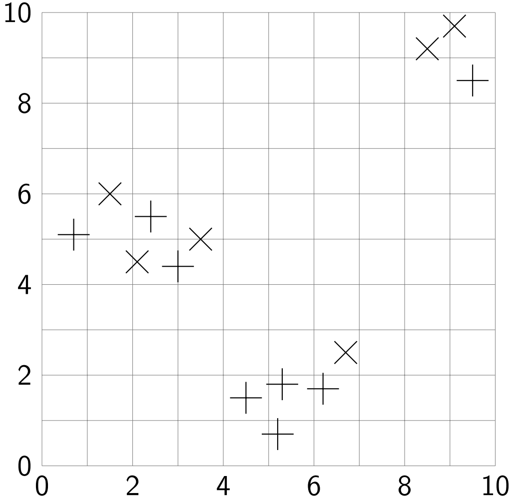
Point	X	Y	Cluster?
P1	0.7	5.1	
P2	1.5	6	
P3	2.1	4.5	
P4	2.4	5.5	
P5	3	4.4	
...	...	...	
P14	9.5	8.5	

Example Data



Number of examples? Number of dimensions?

Random Assignment to TWO Classes

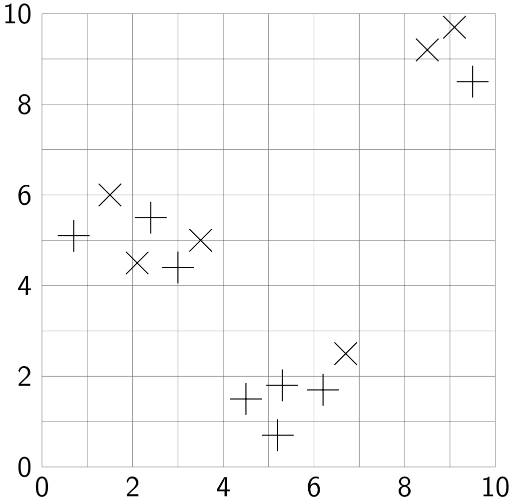


Random Assignment to TWO Classes

As a data set:

Point	X	Y	Cluster?
P1	0.7	5.1	$+$
P2	1.5	6	$\times$
P3	2.1	4.5	$\times$
P4	2.4	5.5	
P5	3	4.4	$+$
...	...	...	
P14	9.5	8.5	$+$

Random Assignment to TWO Classes



Mean of the class $+ :< 4{,}6; 3{,}65 >$; Mean of the class $\times :< 5{,}2; 6{,}15 >$.

Example worked-out

The $+$ class is:

Example

$\underline{(0{,}7;5{,}1)}$, $(1{,}5;6)$, $(2{,}1;4{,}5)$, $\underline{(2{,}4;5{,}5)}$, $\underline{(3;4{,}4)}$, $(3{,}5;5)$, $\underline{(4{,}5;1{,}5)}$, $\underline{(5{,}2;0{,}7)}$, $\underline{(5{,}3;1{,}8)}$, $\underline{(6{,}2;1{,}7)}$, $(6{,}7;2{,}5)$, $(8{,}5;9{,}2)$, $(9{,}1;9{,}7)$, $\underline{(9{,}5;8{,}5)}$.

Example worked-out

The + class is:

Example

$\underline{(0{,}7; 5{,}1)}$, $(1{,}5; 6)$, $(2{,}1; 4{,}5)$, $\underline{(2{,}4; 5{,}5)}$, $\underline{(3; 4{,}4)}$,
$(3{,}5; 5)$, $\underline{(4{,}5; 1{,}5)}$, $\underline{(5{,}2; 0{,}7)}$, $\underline{(5{,}3; 1{,}8)}$, $\underline{(6{,}2; 1{,}7)}$, $(6{,}7; 2{,}5)$,
$(8{,}5; 9{,}2)$, $(9{,}1; 9{,}7)$, $\underline{(9{,}5; 8{,}5)}$.

So the mean for the first feature of this class is:

- Feature X: $\frac{0{,}7+2{,}4+3+4{,}5+5{,}2+5{,}3+6{,}2+9{,}5}{8} = \frac{36{,}8}{8} = 4{,}6$

Example worked-out

The + class is:

Example

$\underline{(0{,}7;5{,}1)}$, $(1{,}5;6)$, $(2{,}1;4{,}5)$, $\underline{(2{,}4;5{,}5)}$, $\underline{(3;4{,}4)}$, $(3{,}5;5)$, $\underline{(4{,}5;1{,}5)}$, $\underline{(5{,}2;0{,}7)}$, $\underline{(5{,}3;1{,}8)}$, $\underline{(6{,}2;1{,}7)}$, $(6{,}7;2{,}5)$, $(8{,}5;9{,}2)$, $(9{,}1;9{,}7)$, $\underline{(9{,}5;8{,}5)}$.

So the mean for the first feature of this class is:

- Feature X: $\frac{0{,}7+2{,}4+3+4{,}5+5{,}2+5{,}3+6{,}2+9{,}5}{8} = \frac{36{,}8}{8} = 4{,}6$

... the mean for the second feature of this class is:

- Feature Y: $\frac{5{,}1+5{,}5+4{,}4+1{,}5+0{,}7+1{,}8+1{,}7+8{,}5}{8} = \frac{29{,}2}{8} = 3{,}65$

Example worked-out

The $\times$ class is in black, not underlined:

Example

(0,7; 5,1), (1,5; 6), (2,1; 4,5), (2,4; 5,5), (3; 4,4),
(3,5; 5), (4,5; 1,5), (5,2; 0,7), (5,3; 1,8), (6,2; 1,7), (6,7; 2,5),
(8,5; 9,2), (9,1; 9,7), (9,5; 8,5).

Doing the same for "black" points will give you mean for $\times$ $< 5{,}2; 6{,}15 >$:

- Feature X: $\frac{1{,}5+2{,}1+3{,}5+6{,}7+8{,}5+9{,}1}{6} = \frac{31{,}4}{6} = 5{,}2$
- Feature Y : $\frac{6+4{,}5+5+2{,}5+9{,}2+9{,}7}{6} = \frac{36{,}9}{6} = 6{,}15$

Computing *pval*

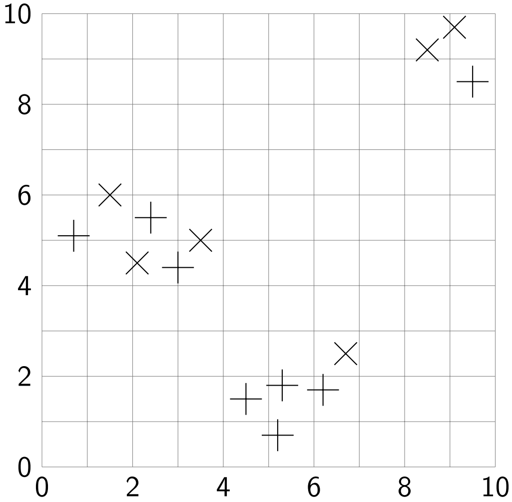
In terms of our algorithm, it is stage **M** that computes:

$$pval(i, X_j) = \frac{\sum_{e:class(e)=i} val(e, X_j)}{|\{e : class(e) = i\}|},$$

$pval(i, X_j)$:

- $pval(+, X) = 4{,}6$
- $pval(+, Y) = 3{,}65$
- $pval(\times, X) = 5{,}2$
- $pval(\times, Y) = 6{,}15$

Lets find them on the picture



Mean of the class $+ :< 4{,}6; 3{,}65 >$; Mean of the class $\times :< 5{,}2; 6{,}15 >$.

Step M: Minimisation of the error

Now lets proceed with step **E**: For each class i,

$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 .$$

<u>We had:</u> $pval$: $+ :< 4{,}6; 3{,}65 >$; $\times :< 5{,}2; 6{,}15 >$
<u>(0,7; 5,1)</u>, (1,5; 6), (2,1; 4,5), <u>(2,4; 5,5)</u>, <u>(3; 4,4)</u>, (3,5; 5), <u>(4,5; 1,5)</u>, <u>(5,2; 0,7)</u>, <u>(5,3; 1,8)</u>, <u>(6,2; 1,7)</u>, (6,7; 2,5), (8,5; 9,2), (9,1; 9,7), <u>(9,5; 8,5)</u>.

Example

- Point 1.
  - $+$:
  - $\times$:

Step M: Minimisation of the error

Now lets proceed with step **E**: For each class i,

$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 .$$

We had: *pval*: $+ :< 4{,}6; 3{,}65 >; \times :< 5{,}2; 6{,}15 >$
(0,7; 5,1), (1,5; 6), (2,1; 4,5), (2,4; 5,5), (3; 4,4),
(3,5; 5), (4,5; 1,5), (5,2; 0,7), (5,3; 1,8), (6,2; 1,7), (6,7; 2,5),
(8,5; 9,2), (9,1; 9,7), (9,5; 8,5).

Example

- Point 1.
  - $+$: $(4{,}6 - 0{,}7)^2 + (3{,}65 - 5{,}1)^2 = 17{,}37$
  - $\times$:

Step M: Minimisation of the error

Now lets proceed with step **E**: For each class i,

$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 \, .$$

We had: $pval$: $+ :< 4{,}6; 3{,}65 >$; $\times :< 5{,}2; 6{,}15 >$
(0,7; 5,1), (1,5; 6), (2,1; 4,5), (2,4; 5,5), (3; 4,4),
(3,5; 5), (4,5; 1,5), (5,2; 0,7), (5,3; 1,8), (6,2; 1,7), (6,7; 2,5),
(8,5; 9,2), (9,1; 9,7), (9,5; 8,5).

Example

- Point 1.
  - $+$: $(4{,}6 - 0{,}7)^2 + (3{,}65 - 5{,}1)^2 = 17{,}37$
  - $\times$: $(5{,}2 - 0{,}7)^2 + (6{,}15 - 5{,}1)^2 = 21{,}35$

Step M: Minimisation of the error

Now lets proceed with step **E**: For each class i,

$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 .$$

<u>We had:</u> *pval*: $+ :< 4{,}6; 3{,}65 >; \ \times :< 5{,}2; 6{,}15 >$
<u>(0,7; 5,1)</u>, (1,5; 6), (2,1; 4,5), <u>(2,4; 5,5)</u>, <u>(3; 4,4)</u>,
(3,5; 5), <u>(4,5; 1,5)</u>, <u>(5,2; 0,7)</u>, <u>(5,3; 1,8)</u>, <u>(6,2; 1,7)</u>, (6,7; 2,5),
(8,5; 9,2), (9,1; 9,7), <u>(9,5; 8,5)</u>.

Example

- Point 1.
  - $+$: $(4{,}6 - 0{,}7)^2 + (3{,}65 - 5{,}1)^2 = 17{,}37$
  - $\times$: $(5{,}2 - 0{,}7)^2 + (6{,}15 - 5{,}1)^2 = 21{,}35$

Which class Point 1 should be assigned to now?

Step M: Minimisation of the error

$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 .$$

We had: $pval$: $+ :< 4{,}6; 3{,}65 >$; $\times :< 5{,}2; 6{,}15 >$
(0,7; 5,1), (1,5; 6), (2,1; 4,5), (2,4; 5,5), (3; 4,4),
(3,5; 5), (4,5; 1,5), (5,2; 0,7), (5,3; 1,8), (6,2; 1,7), (6,7; 2,5),
(8,5; 9,2), (9,1; 9,7), (9,5; 8,5).

Example

- Point 14.
  - $+$: $(4{,}6 - 9{,}5)^2 + (3{,}65 - 8{,}5)^2 = 47{,}73$
  - $\times$: $(5{,}2 - 9{,}5)^2 + (6{,}15 - 8{,}5)^2 = 24{,}01$

Which class Point 14 should be assigned to now?

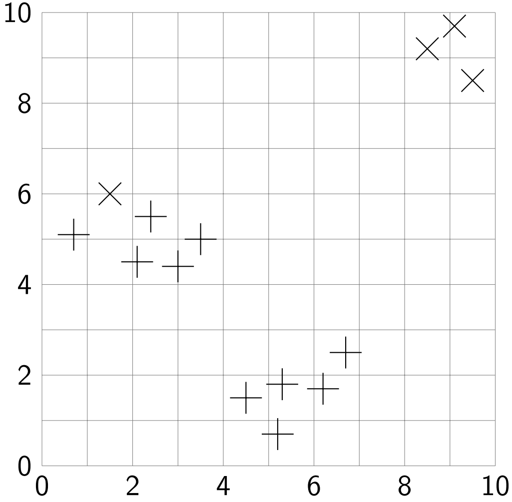
After one iteration,

We had:
$\underline{(0,7;5,1)}$, $(1,5;6)$, $(2,1;4,5)$, $\underline{(2,4;5,5)}$, $\underline{(3;4,4)}$,
$(3,5;5)$, $\underline{(4,5;1,5)}$, $\underline{(5,2;0,7)}$, $\underline{(5,3;1,8)}$, $\underline{(6,2;1,7)}$, $(6,7;2,5)$,
$(8,5;9,2)$, $(9,1;9,7)$, $\underline{(9,5;8,5)}$.

We now have:

$\underline{(0,7;5,1)}$, $(1,5;6)$, $\underline{(2,1;4,5)}$, $\underline{(2,4;5,5)}$, $\underline{(3;4,4)}$,
$\underline{(3,5;5)}$, $\underline{(4,5;1,5)}$, $\underline{(5,2;0,7)}$, $\underline{(5,3;1,8)}$, $\underline{(6,2;1,7)}$, $\underline{(6,7;2,5)}$,
$(8,5;9,2)$, $(9,1;9,7)$, $(9,5;8,5)$.

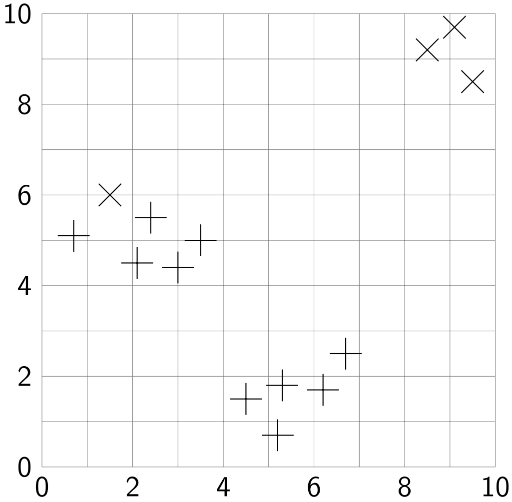
On the plot...



Iteration 2

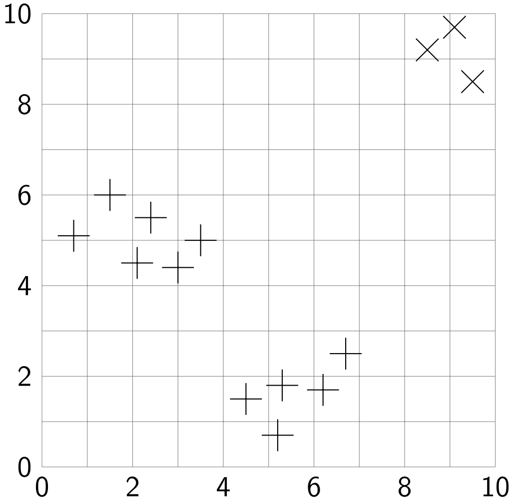
We now repeat the process, and go through the steps **M** and **E** once again...

Assign Each Example to Closest Mean



Mean of the class $+ :< 3{,}96; 3{,}27 >$; Mean of the class $\times :< 7{,}15; 8{,}34 >$.

Ressign Each Example to Closest Mean



This assignment is stable.

The algorithm terminates: what did it LEARN?

Properties of k-means

- An assignment of examples to classes is **stable** if running both the M step and the E step does not change the assignment.

Properties of k-means

- An assignment of examples to classes is **stable** if running both the M step and the E step does not change the assignment.
- This algorithm will eventually converge to a stable local minimum.

Properties of k-means

- An assignment of examples to classes is **stable** if running both the M step and the E step does not change the assignment.
- This algorithm will eventually converge to a stable local minimum.
- Any permutation of the labels of a stable assignment is also a stable assignment.

Properties of k-means

- An assignment of examples to classes is **stable** if running both the M step and the E step does not change the assignment.
- This algorithm will eventually converge to a stable local minimum.
- Any permutation of the labels of a stable assignment is also a stable assignment.
- It is not guaranteed to converge to a global minimum.

Properties of k-means

- An assignment of examples to classes is **stable** if running both the M step and the E step does not change the assignment.
- This algorithm will eventually converge to a stable local minimum.
- Any permutation of the labels of a stable assignment is also a stable assignment.
- It is not guaranteed to converge to a global minimum.
- It is sensitive to the relative scale of the dimensions.

Properties of k-means

- An assignment of examples to classes is **stable** if running both the M step and the E step does not change the assignment.
- This algorithm will eventually converge to a stable local minimum.
- Any permutation of the labels of a stable assignment is also a stable assignment.
- It is not guaranteed to converge to a global minimum.
- It is sensitive to the relative scale of the dimensions.
- Increasing k can always decrease error until k is the number of different examples. (unsupervised form of over-fitting)

The real life scenario for clustering

	X				X		
			X				
	X				X		
		X	X	X			

	X				X		
			X				
X						X	
	X				X		
		X	X	X			

	X				X		
			X				
		X	X	X			
	X				X		

Same data, but there was no-one to mark pictures as happy or sad for us: the AI application has to learn it without our supervision...

Grid face emotions: clustering

It will look like this:

Picture	Cell 33	Cell 42	Cell 48	Cell 58	
P1	White	Black	White	White	
P2	Black	Black	White	White	
P3	White	White	White	Black	
P4	White	White	Black	White	
P5	Black	White	Black	Black	
P6	White	White	Black	Black	
P7	Black	White	White	Black	
P8	Black	White	Black	Black	
P9	White	Black	Black	Black	
P10	White	Black	White	Black	

Unsupervised scenarios

Unsupervised scenarios are common:

- Maybe it is too time-consuming to annotate all data manually
- May be you need to process data on-line, as it comes, and timely manual annotation is impossible
- May be classes are not really known in advance, it is easy enough to tell "happy" from "sad", but not so easy to tell "suspicious" from "trustworthy".

Unsupervised scenarios

Unsupervised scenarios are common:

- Maybe it is too time-consuming to annotate all data manually
- May be you need to process data on-line, as it comes, and timely manual annotation is impossible
- May be classes are not really known in advance, it is easy enough to tell “happy” from “sad”, but not so easy to tell “suspicious” from “trustworthy”.

Research Question

So, is it much harder to find patterns in data when **class is not given**?

Lets find out, in practice: Test 2.

Grid face emotions

Will *k*-means algorithm be able to **restore** the information about happy and sad emotions just by looking at this data set?

Picture	Cell 33	Cell 42	Cell 48	Cell 58	
P1	White	Black	White	White	
P2	Black	Black	White	White	
P3	White	White	White	Black	
P4	White	White	Black	White	
P5	Black	White	Black	Black	
P6	White	White	Black	Black	
P7	Black	White	White	Black	
P8	Black	White	Black	Black	
P9	White	Black	Black	Black	
P10	White	Black	White	Black	

Grid face emotions

Lets simplify its life and remove the confusing feature, Cell 48:

Picture	Cell 33	Cell 42		Cell 58	
P1	White	Black		White	
P2	Black	Black		White	
P3	White	White		Black	
P4	White	White		White	
P5	Black	White		Black	
P6	White	White		Black	
P7	Black	White		Black	
P8	Black	White		Black	
P9	White	Black		Black	
P10	White	Black		Black	

Grid face emotions

Now some entries repeat, and we remove them, as well:

Picture	Cell 33	Cell 42		Cell 58	
P1	White	Black		White	
P2	Black	Black		White	
P3	White	White		Black	
P4	White	White		White	
P5	Black	White		Black	
P9	White	Black		Black	

Grid face emotions

Now some entries repeat, and we remove them, as well:

Picture	Cell 33	Cell 42		Cell 58	
P1	White	Black		White	
P2	Black	Black		White	
P3	White	White		Black	
P4	White	White		White	
P5	Black	White		Black	
P9	White	Black		Black	

Lets use k-means algorithm now!

Grid face emotions

I do some consistent conversion to numeric values:
White to 0, Black to 1.
Starting with stage **E**,

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	
P2	1	1		0	
P3	0	0		1	
P4	0	0		0	
P5	1	0		1	
P9	0	1		1	

Grid face emotions

I do some consistent conversion to numeric values:

White to 0, Black to 1.

Starting with stage **E**, assign randomly all examples to two classes:

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	Sad
P2	1	1		0	
P3	0	0		1	
P4	0	0		0	
P5	1	0		1	
P9	0	1		1	

Grid face emotions

I do some consistent conversion to numeric values:
White to 0, Black to 1.
Starting with stage **E**, assign randomly all examples to two classes:

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	Sad
P2	1	1		0	Happy
P3	0	0		1	
P4	0	0		0	
P5	1	0		1	
P9	0	1		1	

Grid face emotions

I do some consistent conversion to numeric values:

White to 0, Black to 1.

Starting with stage **E**, assign randomly all examples to two classes:

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	Sad
P2	1	1		0	Happy
P3	0	0		1	Sad
P4	0	0		0	
P5	1	0		1	
P9	0	1		1	

Grid face emotions

I do some consistent conversion to numeric values:
White to 0, Black to 1.
Starting with stage **E**, assign randomly all examples to two classes:

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	Sad
P2	1	1		0	Happy
P3	0	0		1	Sad
P4	0	0		0	Happy
P5	1	0		1	
P9	0	1		1	

Grid face emotions

I do some consistent conversion to numeric values:

White to 0, Black to 1.

Starting with stage **E**, assign randomly all examples to two classes:

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	Sad
P2	1	1		0	Happy
P3	0	0		1	Sad
P4	0	0		0	Happy
P5	1	0		1	Sad
P9	0	1		1	

Grid face emotions

I do some consistent conversion to numeric values:
White to 0, Black to 1.
Starting with stage **E**, assign randomly all examples to two classes:

Picture	Cell 33	Cell 42		Cell 58	Cluster
P1	0	1		0	Sad
P2	1	1		0	Happy
P3	0	0		1	Sad
P4	0	0		0	Happy
P5	1	0		1	Sad
P9	0	1		1	Happy

Homework: Test 2 Part 1

Take this data set and this random assignment

Manually execute k-means algorithm on it until it converges, be ready to answer my questions about your intermediate computations as well as the final results. Compare the results to the class labels given in Bayes1.pdf: were they recovered by clustering?
Convention: when re-assigning classes, if same distance is computed for both classes, give preference to Sad.

Homework: Test 2 Part 1

Take this data set and this random assignment

Manually execute *k*-means algorithm on it until it converges, be ready to answer my questions about your intermediate computations as well as the final results. Compare the results to the class labels given in Bayes1.pdf: were they recovered by clustering? Convention: when re-assigning classes, if same distance is computed for both classes, give preference to Sad.

Reading:

- Check relevant chapters on Clustering in the recommended textbook: Data Mining, by Witten et al. (2011) §6.8 (pp 273-294), §11.6 (pp.480-485).
- In 2017 edition: §4.8 (pp. 141 – 156), on-line appendix `https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf`: §2.5 (pp. 43-44)

Tomorrow

- We will take our knowledge of k-means algorithm and Bayesian learning to new hights:
  - We will combine them in a soft clustering algorithm